DEVELOPMENT OF TWO COOPERATIVE STEWART PLATFORMS FOR MACHINING

VINCENSIUS BILLY SAPUTRA

B.Eng., ITB

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2012

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Vincensius Billy Saputra

27 September 2013

ACKNOWLEDGEMENTS

I am deeply indebted to my supervisors, Prof. Andrew Nee Yeh Ching, and Assoc. Prof. Ong Soh Khim for their suggestions, guidance, insights, and patience that have been invaluable to this research project and this thesis, and will long be treasured and greatly appreciated.

I would like to thank all fellow students in our research group and AR group, especially Dr Ng Chee Chung, Dr Niu Sihong, Dr Fang Hongchao, for their friendship, help, encouragement, research ideas and opinions. In addition, I am very much obliged to the Advanced Manufacturing Laboratory for their assistance throughout the research. I express my gratitude to the Department and laboratory staff members, especially Mr Tan Choon Huat, Mr Lim Soon Cheong, Mr Ho Yan Chee, Mr Wong Chian Long, Mr Simon Tan Suan Beng, and Mr Lee Chiang Soon, for their administrative and technical help with the project.

I appreciate the work done by FYP, undergraduates and JC students that in many ways help this research project and the group. I would especially thank Low Minyi Cindy, Chock E-Wei, V.M. Ajay, and Ye Chenhao for their contribution. Finally, I would like to thank my family and my friends for their support and encouragement.

TABLE OF CONTENT

DECLARATION PAGE ..... **I**

ACKNOWLEDGEMENTS ..... **II**

TABLE OF CONTENT ..... **III**

SUMMARY ..... **VII**

LIST OF TABLES ..... **IX**

LIST OF FIGURES ..... **X**

LIST OF ABBREVIATIONS ..... **XIII**

LIST OF SYMBOLS ..... **XIV**

CHAPTER 1 INTRODUCTION **1**

1.1 Overview **1**

1.2 Background **2**

1.2.1 Serial Architecture **2**

1.2.2 Parallel Architecture **3**

1.2.3 Hybrid Architecture **4**

1.3 Organization **6**

1.4 Objectives of the Study **8**

CHAPTER 2 LITERATURE REVIEW **10**

2.1 Kinematics 10

2.2 Workspace and Singularities 11

2.3 Calibration and Accuracy 14

2.4 Motion Planning and Redundancies 16

2.5 Dynamics and Control 18

2.6 Stewart Platform for Machining Applications 20

CHAPTER 3 COMPUTER NUMERICALLY CONTROL MACHINE TOOL CONCEPTS 23

3.1 Part Geometry Design 25

3.2 PKM-based Machine Tool Advantages 27

CHAPTER 4 THE COOPERATIVE MANIPULATORS DESIGN AND IMPLEMENTATION 29

4.1 Cooperative Manipulators Structure Description 29

4.2 Coordinate Systems and Kinematics 32

4.3 Components and Control System 35

4.3.1 Tool Stewart Platform 35

4.3.2 Table Stewart Platform 38

4.3.3 Design Consideration 41

4.3.4 Joints Location 45

4.3.5 Frame Design 48

4.4 Single Stewart Platform Configuration 48

4.5 Extended Configuration 51

CHAPTER 5 SIMULATION AND CONTROL OF STEWART PLATFORM 53

5.1 Workspace Analysis and Kinematic Constraints 53

5.2 Stewart Platform User Interface 58

5.3 Programming 62

5.4 Numerical Control Post-Processor for Stewart Platform 68

5.5 Stewart Platform Motion Emulation and Dynamics 71

CHAPTER 6 IMPLEMENTATION OF THE COOPERATIVE MANIPULATORS AS MACHINE TOOL 73

6.1 Coordinate Mapping of the Cooperative Manipulators 73

6.1.1 Single Stewart Platform Configuration 74

6.1.2 Extended Configuration 80

6.2 Extended Configuration Motion Planning 88

6.2.1 Jacobian Matrix and Condition Number 89

6.2.2 Optimization Procedure 92

6.2.3 Straight-line Milling 94

6.3 Stewart Platform Machining Framework with CAD/CAM Software 98

6.3.1 Tool Path Post-processing 102

6.3.2 Determining Machine Origin 106

6.4 Machining Case Studies 107

6.4.1 Machining an 'NUS' Pocket 107

6.4.2 Machining a Dome 109

6.4.3 Machining a Test Part 113

6.4.4 Machining with Rotation Axes 117

CHAPTER 7 STEWART PLATFORM MACHINING OPTIMIZATION AND EVALUATION 120

7.1 Machining Workspace Analysis 120

7.2 Application of workspace data for optimal setup in machining 124

7.3 Calibration and Accuracy Improvement 125

7.3.1 Perpendicularity of Dial Gauges 134

7.3.2 Pose selection for Calibration 137

7.3.3 Online Calibration for Kinematic Parameters Error Compensation 142

7.4 Machining Evaluation 144

7.5 Stewart Platforms Evaluation 151

CHAPTER 8 CONCLUSIONS AND RECOMMENDATIONS 154

8.1 Conclusions 154

8.2 Research Contributions 156

8.3 Future Work 158

BIBLIOGRAPHY 160

PUBLICATIONS FROM THIS RESEARCH 177

SUMMARY

While very large and heavy duty machines are still needed for high volume mass production, there is a growing need in today's manufacturing for lighter production machines with smaller size and mass to increase the efficiency in certain sectors that produce low volume customized products. This research investigates the application of Parallel Kinematic Manipulators (PKM), namely Stewart platforms, for such manufacturing applications especially for machining and positioning. PKMs have inherent properties for machining applications, but the main constraint of PKMs is the limited workspace. In this study, cooperative manipulators comprising a configuration of two Stewart platforms is built. The two Stewart platforms interact with one another. One of them carries the tool and the other one holds the object. This approach increases the flexibility of the cooperative manipulators to handle multi-axis machining jobs and enables the cooperative manipulators to achieve larger workspace and wider tilting ranges.

The scope of this research includes the modelling of the Stewart platforms, design methodology for optimal geometric parameters, test of a prototype for error compensation and an analysis of the machining results. The motion control input is implemented with translation from standard G-codes such that a commercial CNC software can be used. An optimization strategy is developed to solve extra degrees of freedom with objectives related

to the characteristics of the Stewart platforms. Development and results of the cooperative manipulators is presented.

LIST OF TABLES

Table 4.1 Search range for the dimensional synthesis 44

Table 4.2 Stewart platform joint locations (in mm) 46

Table 5.1 NC codes and their functions 69

Table 5.2 G-codes used for Stewart platform and their meaning 71

Table 6.1 Values used in the example 95

Table 6.2 Trajectory Planning Result Summary 95

Table 7.1 Workspace Volume with various tilt angles of the tool axis 124

Table 7.2 Real model for calibration simulation 133

Table 7.3 Error comparison of calibration simulation 133

Table 7.4 Kinematic Parameters after Calibration 134

Table 7.5 Error comparison of calibration with measurement errors 136

LIST OF FIGURES

Figure 1.1 The first octahedral hexapod or the Gough Stewart platform 4

Figure 1.2 The Logabex robot LX4 (courtesy of Logabex Company) 6

Figure 1.3 Operational model of hybrid robotic arm 6

Figure 4.1 Two Stewart platforms 31

Figure 4.2 Extended Configuration of the cooperative manipulators 32

Figure 4.3 Schematic representation of the Stewart platform 33

Figure 4.4 Tool-SP (not installed in the frame) 36

Figure 4.5 Actuator and passive joints in the tool-SP 36

Figure 4.6 The tool attached to the moving platform 37

Figure 4.7 The PC controller card for the tool-SP 38

Figure 4.8 The table-SP (outside the frame) 40

Figure 4.9 The passive joints of the table-SP 41

Figure 4.10 Relation ship between joints positions 43

Figure 4.11 Dimensional parameter synthesis of Stewart Platform 45

Figure 4.12 Joints location of the tool-SP 46

Figure 4.14 Single Tool Stewart platform Configuration 50

Figure 4.15 The machining table installed in the frame 50

Figure 4.16 Machining a clamped work-piece on the table-SP with the tool-SP 52

Figure 5.1 The flowchart to plot the workspace of the Stewart platform 57

Figure 5.2 The tool Stewart platform workspace..... 58

Figure 5.3 User Interface for Stewart platform control..... 60

Figure 5.4 Point-to-point motion control flowchart..... 65

Figure 5.5 Spline motion control flowchart 66

Figure 5.6 Jog motion control flowchart..... 67

Figure 5.7 Interpolated motion control flowchart 68

Figure 5.8 Common NC program format..... 69

Figure 5.9 A 3D Stewart platform model..... 72

Figure 5.10 Stewart platform model in MatLab Simulink..... 72

Figure 6.1 Coordinate system in the single configuration 75

Figure 6.2 The moving platform position to reach the tool contact point.... 79

Figure 6.3 Coordinate mapping in single configuration 80

Figure 6.4 Coordinate system in extended configuration 83

Figure 6.5 Coordination of the tool-SP and the table-SP to reach the input coordinate (the tool contact point on the work-piece) 84

Figure 6.6 Rotation sequence in the tool-SP..... 87

Figure 6.7 Rotation sequence in the table-SP 87

Figure 6.8 Coordinate mapping in extended configuration 88

Figure 6.9 Algorithm for motion planning in extended configuration..... 94

Figure 6.10 Resulting trajectory plan q_z from the optimization procedure.. 96

Figure 6.11 A smooth trajectory from the algorithm with extra constraint; showing: 98

Figure 6.12 Steps to machining with the cooperative manipulators 100

Figure 6.13 Fixture for holding work-piece on top of the tool-SP 100

Figure 6.14 Information flow of part design and NC code generation 102

Figure 6.15 Machining Input Parameters..... 102

Figure 6.16 Additional information in a G-Code file that cannot be processed by the MatLab post processor (blue codes)..... 103

Figure 6.17 Block Numbers removal in the NC code..... 104

Figure 6.18 Testing feasibility of the resulting trajectory, (a) Inaccessible trajectory (b) Partially accessible trajectory (c) Accessible trajectory..... 105

Figure 6.19 Determining the machine origin 107

LIST OF ABBREVIATIONS

PKM	Parallel Kinematic Manipulator
SP	Stewart platform
DOF	Degree Of Freedom
CNC	Computer Numerically Control
LSF	Least Squares Formulation
TLSF	Total Least Squares Formulation
PD	Proportional Derivative
PID	Proportional Integral Derivative
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
FKM	Forward Kinematic Model
IKM	Inverse Kinematic Model
CMM	Coordinate Measuring Machine
PSO	Particle Swarm Optimization
CC	Cutter Contact
CL	Cutter Location

LIST OF SYMBOLS

$X_P, Y_P Z_P$	Coordinate axes of the Stewart platform moving platform
$X_F, Y_F Z_F$	Coordinate axes of the Stewart platform base
a_i	Platform attachment joints, ball-socket joints, i = 1,2…6
b_i	Base attachment joints, universal joints, i = 1,2…6
$\{F\}$	Coordinate frame attached to the Stewart platform base
$\{P\}$	Coordinate frame attached to the moving platform
$\{W\}$	Coordinate frame attached to the machining table
l_i	Leg vector
$\underline{L_i}$	The ith leg length
L_{Oi}	The ith leg offset
$\underline{\boldsymbol{R}}$	Rotation matrix
q	The position vector of $\{P\}$ relative to $\{F\}$
θ	The vector of orientation angle of $\{P\}$ relative to $\{F\}$
S	Sine
C	Cosine
$\dot{\boldsymbol{X}}$	Vector of Cartesian velocities of the moving platform
$\dot{\boldsymbol{l}}$	Vector of leg velocities
$\boldsymbol{X}$	Vector of generalized coordinates of the moving platform
$\boldsymbol{X}_P$	A pose set for calibration
Y	Measured variables
η	A vector containing all kinematic parameters

F_i	The objective function for the i^{th} pose
Q	A vector of computed nominal leg length
$\boldsymbol{J}$	The Stewart platform Jacobian matrix
$\boldsymbol{J}_P$	Identification Jacobian matrix
O	Observability index (Noise Amplification Index) calculated from $\boldsymbol{J}_P$
σ	Singular values matrix $\boldsymbol{J}_P$
κ	Condition Number
R_a	Radius of the moving platform of the SP
R_b	Radius of the base of the SP
α	Angle between two closest joints on the moving platform of SP
β	Angle between two closest joints on the base of SP

CHAPTER 1

INTRODUCTION

1.1 Overview

Currently, robots are applied in a variety of manufacturing applications, such as machining, welding, polishing, assembly, pick and place, etc. This has triggered the accelerated development and applications of robotic manipulators in manufacturing. There are two common classes of industrial applications of robotic manipulators, namely, serial and parallel kinematic manipulators. Serial manipulators have open kinematic chain and parallel kinematic manipulators (PKMs) have closed structure of links and joints. This thesis improves upon the development of a specific type of PKMs, which are also known as Stewart platforms (SP) for positioning and machining applications, and investigates the control and user interface aspects.

PKMs have not received much attention as compared to the serial counterparts probably because of the complexities due to their limited workspace, control and their singularity characteristics, which sometimes can occur within the workspace and need to be avoided at all cost. Such problems are seldom found in conventional industrial serial manipulators. Nevertheless, this does not stop researchers from developing new strategies to work with PKMs and come up with new methods that could bring PKMs to their full potential. Therefore, this research proposes a combined structure or a cooperative architecture that consists of two PKMs to study the effect of extra degrees-of-freedom (DOFs) and investigate whether this cooperative

architecture could bring additional benefits to the applications of PKMs in manufacturing and other industrial fields.

1.2 Background

The invention of the first robotic manipulator has triggered development in many research and industry fields, such as satellite positioning, underwater explorations, medical operations, flight simulators, etc. However, there are actually three basic robot architectures, namely as follows:

1. Serial architecture
2. Parallel architecture (PKMs)
3. Hybrid architecture

These three architectures are classified based on the basic structure of a robot's kinematic properties, which are closely related to the sequence and the arrangement of the joints and links in a robot manipulator.

1.2.1 Serial Architecture

Serial manipulators usually consist of at least two and up to a maximum of eight rigid links and joints with some prismatic and revolute joints which can be passive and active (actuated). The main advantage is their large workspace resulted from their first long links from the base and followed by wrists with three or fewer DOFs connected to an end-effector or a tool suitable for a specific task. The serial architecture also tends to have higher dexterity. Their weaknesses are mainly their limited payload, low

precision due to the bending force over the long links connected in a serial manner, and the large number of parts leading to high inertia which is undesirable for high bandwidth motion control. The high inertia disadvantage prevents the use of serial robots for applications requiring high accelerations and agility, e.g., flight simulation and rapid pick and place tasks.

1.2.2 Parallel Architecture

Parallel kinematic manipulators are built from a series of closed kinematic chains. A Stewart platform or Gough-Stewart platform, the first PKM, is composed of six variable struts that are driven by prismatic actuators, connected to a fixed rigid body and a moving platform which position and orientation can be changed based on the lengths of the struts, as shown in Figure 1.1 (Gough and Whitehall 1962). Stewart suggested that the structure be used for high payload applications because every actuator is located to the base, reducing the inertia (Stewart 1965). Compared to serial manipulators, PKMs have higher payload-to-weight ratio, higher stiffness, and higher precision due to their structure in which errors in each link do not add up to be transferred to the end-effector. On the other hand, the disadvantages of the PKMs include difficulty in the control strategies, complicated direct kinematics, inconsistent performance over the workspace, and the occurrence of singular configurations.

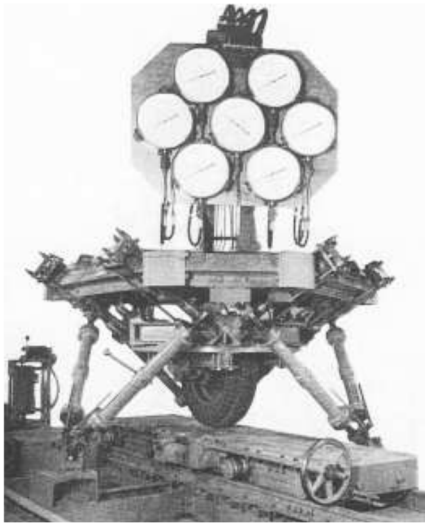


Figure 1.1 The first octahedral hexapod or the Gough Stewart platform

1.2.3 Hybrid Architecture

To overcome those problems discussed in the last section, researchers have explored the combination of serial and parallel structures to form hybrid structures, in order to combine the advantages of serial and parallel structures and complement the drawbacks of each structure. Promising results have been reported, e.g., the Logabex LX4 robot (Figure 1.2) or the robotic arm designed at the California Institute of Technology (Figure 1.3) (Tanev 2000). These manipulators consist of identical parallel mechanisms piled up, and possess a large workspace and a good ratio of load capacity/manipulator mass. Furthermore, some hybrid manipulators have already been used in applications, e.g., deep-sea mining, machining, and medical and assembly operations (Chai and Young 2001; Callegari and Suardi 2003; Zheng *et al*. 2004; Carbone and Ceccarelli 2005; Harib *et al*. 2007). Hybrid structures are usually designed such that one platform performs pure translation and the other performs orientation so as to simplify the control algorithm (Lallemand

et al. 1997; Tsai and Joshi 2002). However, it is still uncertain whether this option will solve the problems and bring forth the full potential of both types of structures into practice. Therefore, this research may contribute to the literature on this class of hybrid manipulators.

A study on the performance comparison between serial and parallel structures has been conducted (Geldart *et al.* 2003). The result showed that the particular parallel kinematic manipulators outperformed the conventional machining centres while cutting hard material. In addition, a comparison of variations of Gough Stewart platform can be found in the literature (Weck and Staimer 2002; Schwaar *et al*. 2002), of which some of the structures comprise hybrid architecture. Although some conceptual and practical industrial works have been already done for the hybrid architecture, this thesis focuses on a cooperative scheme of two PKMs. There are several successes in the past where variants of PKMs are developed for specific applications (Terrier *et al.* 2005; Refaat *et al*. 2007; Neuman 2006). The Gough Stewart platform manipulator is selected in this research, because it is simpler to build with modular components, and therefore can lead to lower cost.

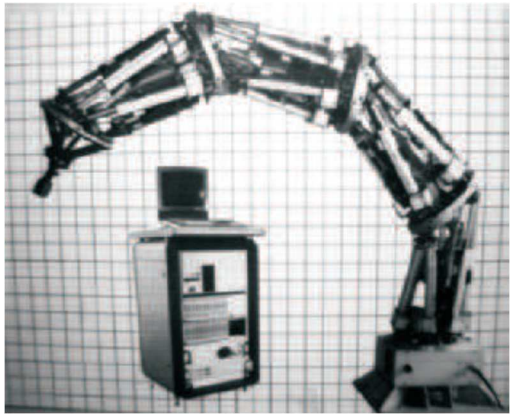


Figure 1.2 The Logabex robot LX4 (courtesy of Logabex Company)

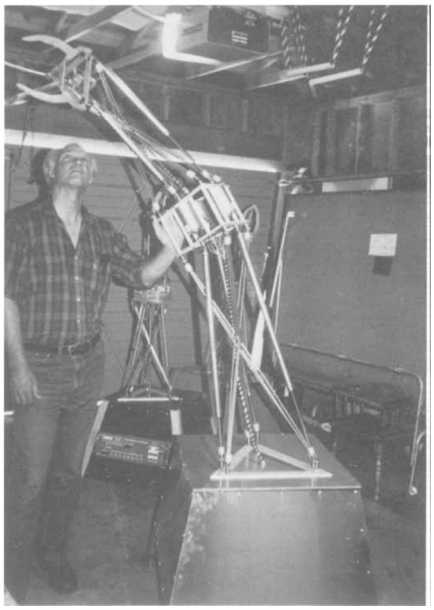


Figure 1.3 Operational model of hybrid robotic arm

1.3 Organization

Stewart platform mechanisms are less intuitive to evaluate than conventional serial mechanisms. Design and analysis must be performed using models and simulation tools. The research on the Stewart platforms in this thesis is addressed in several phases. First, a kinematic model and 3D solid model are built to analyse the motion of the Stewart platforms. Second, a post-processor to translate machining tool paths obtained from CAD/CAM into the Stewart platform trajectory is developed. Third, basic and complex programs are run to test the Stewart platforms, where parts are designed and fabricated. Lastly, tools and methods for process planning, machine simulation, operation, calibration and redesign are explored.

Since there have been much research on several aspects of PKMs, there is a considerable amount of background literature on these topics. Chapter 2 is devoted to a brief review of the concepts and results from relevant literature. Chapter 3 introduces computer numerically control (CNC) concepts that are used in relation to the configuration of the Stewart platform as a machine tool. It reviews the general process by which ordinary CNC machines are operated and the functional requirements of the Stewart platform as a machine tool.

Chapter 4 focuses on aspects related to the hardware of the Stewart platforms being investigated in this research work. Chapter 5 presents the software aspect of the Stewart platforms including the simulation and computation tools developed for controlling the Stewart platforms, providing graphical interface and characteristic analysis that are useful for future design.

Chapter 6 explains the crux of the motion planning algorithm for the proposed configuration with a commercial CAD/CAM system in order to operate the Stewart platforms to execute various machining tasks. This chapter explains how redundancy introduced in the cooperative manipulators with two Stewart platforms can be used to plan the optimal motion path for a given tool path trajectory. In addition, machining case studies which have been executed with the proposed Stewart platform are presented. From these cases, comparisons are made based on the machining results of a single Stewart platform and the cooperative configuration consisting of two Stewart platforms. Observations are made based on the experiments performed and

are used to evaluate tool path generation, work-piece setup, user experience, and future design considerations.

In Chapter 7, several improvements in the workspace and accuracy of the Stewart platform are reported. The error model and calibration of the Stewart platforms to compensate the inaccuracies caused by assembly and manufacturing errors is presented.

Chapter 8 concludes the investigation of the Stewart platforms configuration and application. It also summarizes the results and suggests areas where further work is recommended.

1.4 Objectives of the Study

Stewart platforms have several potential applications. With respect to this research, one of their uses is in a flexible manufacturing environment. In principle, the end-effector can be positioned in any way that is required for the respective task, e.g., milling, welding, cutting and assembly. In particular, the goals of this thesis are:

1. To investigate the integration of two Stewart platforms (six DOF and three DOF) or (six DOF and six DOF) to form a nine or twelve DOF system. For example, one of the SPs can be used to locate and hold a work-piece, and another Stewart platform can be used to hold a cutting tool or some other measuring devices.
2. To explore the development of user interfaces that can be used to control two Stewart platforms simultaneously to plan for the necessary machining paths and avoid any collision. The two Stewart

platforms can move together such that the time to reach the final position will be shortened.

3. To obtain the work volume of the two Stewart platforms and provide calibration and feedback control of the two coordinated Stewart platforms to compensate for any inaccuracies in movements and final positions.
4. To carry out case studies to study multi-axis machining operations. Due to the restriction of the movements of the coupled SP system, it will be necessary to explore the type of work-piece geometry that can be handled in a single set-up. A sub-objective here is to explore, given a particular work-piece model in 3D, the accessible and inaccessible features in a particular set-up and optimize the orientation of the part such that the total number of set-ups required is a minimized.

CHAPTER 2

LITERATURE REVIEW

Most of the research on PKMs deals with conventional robotic issues, such as kinematics, singularities, dynamics, workspace, calibration accuracy and structural properties (Merlet 1999; Dasgupta and Mruthyunjaya 2000). Few research studies have been reported on motion planning, control and robot design or synthesis. Hybrid manipulators have received attention in these areas (Zhang *et al*. 2005). In this chapter, some key issues in this field will be reviewed, although not exhaustively.

2.1 Kinematics

The kinematics of SP mechanisms, like all robotic manipulators in general, is a study of the geometry of the motions of the end-effector and the actuating joints, and the relationship between these two types of motions without consideration of the torques and forces that cause these motions. The inverse kinematics problem, i.e., to find the lengths of the links for a given position of the moving platform, is quite straightforward for PKMs. On the contrary, direct kinematic problem has to be solved using numerical methods. For a general SP, 40 assembly modes (i.e., direct model solutions) can exist (Dietmaier 1998). In practice, the use of numerical procedures has been proposed, which assume that an estimated solution is known (Nguyen *et al*.

1991; Parikh and Lam 2005; Wang 2006). Another method is to use a larger number of sensors than the number of DOFs so that additional information can be used to improve the direct kinematics algorithm (Cheok *et al*. 1993; Parenti-Castelli and Di Gregorio 1995; 2000; Chen and Fu 2006). It has been shown that the computation of forward kinematics is more efficient with an additional off-line pre-processing phase (Tarokh 2007).

2.2 Workspace and Singularities

Singular configurations are particular poses of the end-effector or moving platform of the PKM for which the manipulator loses its inherent rigidity, and the end-effector has uncontrollable degrees of freedom. At singularity positions, the joint velocities may be unbounded although the linear velocity and the angular velocity of the robot arm are bounded. The occurrence of singular configurations is highly undesirable in PKMs since in these configurations, the actuators cannot control the mechanisms, which gain additional finite or infinitesimal freedom and variation in stiffness (Merlet 1992). There are two Jacobian matrices $\boldsymbol{J}$ for a PKM (Gosselin and Angeles 1990b), i.e., one for the inverse kinematic and one for the direct kinematic. This yields three types of singularities:

1. $\det\left(\boldsymbol{J}_{dir.kin.}\right)=0 \quad \text{and} \quad \det\left(\boldsymbol{J}_{inv.kin.}\right)\neq 0$
2. $\det\left(\boldsymbol{J}_{inv.kin.}\right)=0 \quad \text{and} \quad \det\left(\boldsymbol{J}_{dir.kin.}\right)\neq 0$
3. $\det\left(\boldsymbol{J}_{dir.kin.}\right)=0 \quad \text{and} \quad \det\left(\boldsymbol{J}_{inv.kin.}\right)=0$

Depending on which of the two matrices are singular, a PKM may be at a direct kinematic singular configuration, an inverse kinematic singular

configuration, or both. In addition, another type of singularity exists, namely, architecture singularities (Ma and Angeles 1991a). This type of singularity arises from the symmetrical architecture of the SPs causing singularity poses over a significant portion of the entire workspace. Some research focuses on the characterization of the singularity of SPs (Hunt 1978; Fichter 1986; St-Onge and Gosselin 2000), such as detecting singularities in a given workspace (Merlet 2007), and a numerical procedure for avoiding singularities of a SP by restructuring a pre-planned path in the vicinity of a singularity (Bhattacharya *et al*. 1998). In addition, a more recent method for measuring closeness to singularity is by using physical properties, such as the stiffness and torque transmission of the PKM, which better capture all the singularity configurations (Voglewede and Ebert-Uphoff 2005). Furthermore, a more complete analysis of singularity configurations can be done through acknowledging that they are configurations of singularities that cannot be simply detected by computing the Jacobian matrix. This type can be detected only if proper input-output velocity analysis is used (Zlatanov *et al*. 2002; Han *et al*. 2002).

A main drawback of PKMs is their limited workspace. There are three main mechanical constraints that restrict the workspace of PKMs, specifically, the actuators' stroke, the range of the passive joints, and the link interference (Bonev and Ryu 2001). The workspace of a manipulator is defined as the set of all the end-effector configurations that can be reached. Various methods to determine the workspace of a PKM have been proposed using geometric or numerical approaches. Since the translational and orientation workspaces of

a SP are coupled, a first approach is to fix the values of a few of the DOFs until only three DOF are free, so that it can be represented in a 3D plot. Most often, the 3D constant orientation workspace, which describes the possible location of the origin of the end-effector for a constant orientation, is of interest. A geometrical approach (Gosselin 1990b) that has been reported gives the best result as it provides an exact calculation with a compact storage and easy representation. However, numerical methods (Du Plessis *et al*. 2001) are also preferred as they can deal with joint limits and workspace verification problems (Masory and Wang 1995) (i.e., to determine whether a part of the workspace is reachable).

One application of workspace analysis lies in the field of machine tools, where only five DOFs are required for completing a task. Workspace analysis for PKM based machine tools has been reported (Wang *et al*. 2001). Huang (Huang *et al*. 1999) showed that the minimum reachable yaw angle for a given point may be calculated exactly when the constraints on the passive joints are modelled using a cone. Another aspect of workspace analysis for machine tools is part positioning, i.e., given a machining operation to be performed, the problem is to determine the positioning of the part that the machining trajectory will lie within the workspace. This problem has been addressed using the discretization approach (Pugazhenthi *et al*. 2002).

2.3 Calibration and Accuracy

PKMs were introduced due to their higher accuracy as compared to conventional robots and better stiffness in the same range as the machine tools. Due to the complicated kinematic chains in a PKM, it is difficult to achieve the required accuracy. Generally, the error sources of a SP-based machine tool can be classified into geometric and non-geometric. Geometric errors are errors in the parameters that define the geometric relationships. Other sources of errors are grouped as non-geometric errors. In the accuracy or error analysis, it is necessary to develop a valid error model through examining the main sources of errors and investigate the relationships between the errors of the joints and those of the end-effector. Masory (Masory *et al*. 1997) has studied the influence of the sensor errors and the manufacturing tolerances on the locations of the joint centres. A more thorough analysis has been proposed by Ehmann *et al*. (Petal and Ehmann 1997; Wang and Ehmann 2002), which includes the location errors of the passive joint centres, errors in the leg lengths, and the imperfect motions of the ball joints. Tischler and Samuel (Tischler and Samuel 1998) proposed a numerical approach for determining the influence of the backlash of the joints, while Meng (Meng and Li 2005) and Wolhart (Wohlhart 1999) proposed an analysis of the effect of the joint clearances on the trajectories followed by serial and parallel manipulators. Other sources of errors, such as thermal errors, gravity induced errors, and dynamic errors (Pritschow *et al*. 2002, Niaritsiry *et al*. 2004; Clavel 2005), have also been studied.

Geometric errors, sometimes called the kinematic errors, can be reduced through kinematic calibration, which deals with the improvement of a kinematic model of a manipulator that is attainable through substituting the nominal values of the kinematic parameters with their actual values. In kinematic calibration, various methods have been suggested, e.g., optimization methods (Zhuang and Roth 1993), linearization method (Geng and Haynes 1994), and partial differentiation (Ropponen and Arai 1995). Merlet (Merlet 2006) distinguished three main types of calibration methods:

1. ***External calibration***: An external measurement device is used to determine (completely or partially) the real pose of the platform for different desired configurations of the moving platform. The differences between the measured pose and the desired pose give an error signal that is used for the calibration
2. ***Constrained calibration***: Methods that rely on a dedicated mechanical system that constrains the robot motion during the calibration process.
3. ***Auto-calibration or self-calibration***: The platform has extra sensors and only the manipulator measurements are used for the calibration. In this case, it is required that an n-DOF robot has m (> n) internal sensors.

In addition, there is another group of calibration methods which uses interesting geometrical properties. Huang (Huang *et al*. 2005) proposed using specific motion characteristics, e.g., flatness and straightness which can be measured easily using dial gauges. Takeda (Takeda *et al*. 2004) proposed

using a double-ball-bar measuring device. In his study, the robot performed circular paths, and the deviation from circularity was measured using the device. In the machining field, calibration can be conducted using machining experiments (Chanal *et al*. 2007). Recent research shows a trend using camera calibration that can produce good accuracy with relatively low cost (Andreff *et al*. 2004; Dallej *et al*. 2006a; Daney 2006; Renaud *et al*. 2006; Tanaka 2006).

2.4 Motion Planning and Redundancies

Motion planning is a classical problem for serial manipulators to avoid obstacles. However, for parallel manipulators, more factors have to be considered, such as limited workspace, singularities, and other performance requirements. Merlet (Merlet 1994) presented a method for checking whether a trajectory lies within the workspace of a manipulator. Harris (Harris 1995) dealt with motion planning between two poses by looking for the parameters of the screw motion linking the two poses, and reckoning that this motion should be able to minimize the changes in the link lengths. Gosselin and Angeles (Gosselin and Angeles 1990c) presented an algorithm that can find the orientation of the manipulator with the best accuracy in some specific poses along a path based on the condition number. Recently, probabilistic path planning has emerged as one of the most promising approaches to path planning of manipulators with large DOFs. A most prominent research in this field for parallel manipulators is the probabilistic roadmap approach (Cortes and Simeon 2003). However, this approach does not consider singularity or

multiple solutions for the direct kinematics, which may prohibit the use of the trajectory.

Motion planning for machine tools may present some specificities, as fewer than the number of DOFs of the machine may be used, e.g., for a SP, the rotation about the tool can be ignored as it does not have to be specified for machining tasks. Therefore, it is possible to determine the ranges for the free DOFs to ensure that a given machining trajectory lies within the workspace and apply an optimization procedure on the free DOFs to optimize other performance criteria for the SP (Merlet 2000). Another approach (Chen *et al*. 2003) partitioned the DOFs into critical and secondary DOFs, and synthesized a control law that ensured the tracking of the critical DOFs while minimizing a velocity-based secondary criterion.

Redundant manipulators are of significant importance because of their advantages when task versatility and manipulator performances are required. Non-redundant manipulators, serial or parallel, perform well over a certain range of task operations corresponding to the limitations of their structural and actuation characteristics. Redundant manipulators possess 'additional inputs' that offer a means to improve their performance and increase their versatility. Pierrot (Pierrot 2002) distinguished three different types of redundancies:

1. ***Kinematic redundancy***: At least one of the legs is a motion generator with a larger number of DOFs than necessary. This may be used for enlarging the workspace (Liu *et al*. 2001).

2. ***Actuation redundancy***: The end-effector is over constrained by the actuators. The number of actuators is more than the number of DOFs. Such redundancy is mostly used for singularity avoidance (Wang and Gosselin 2004).
3. ***Measurement redundancy***: The number of sensors is larger than the number of actuated joints. This redundancy plays a role in solving the forward kinematic problem to reduce the positioning errors and for calibration (Marquet *et al*. 2002).

When two SPs are combined, there is one redundancy caused by the two SPs working together simultaneously. Thus, the main problem is to determine an optimum use of this redundancy. A machining path for cooperative manipulators consisting of these two SPs can be generated only when redundancy has been resolved.

2.5 Dynamics and Control

Dynamics is the determination of the relationship between the generalized accelerations, velocities, and coordinates of the end-effector and the joint forces. Dynamic analysis of PKMs is complicated by the existence of multiple closed-loop chains. The earliest discussion on computing the dynamics of SPs can be found in research work by Fichter (Fichter 1986), which is applicable when the leg inertia and the joint friction are negligible. As the SPs became better known, there were three major approaches of computing the dynamics, namely, the Newton-Euler formulation (Codourey and Burdet 1997; Dasgupta and Choudhury 1999; Harib and Srinivasan 2003),

the Lagrangian formulation (Nguyen and Pooran 1989; Geng *et al*. 1992; Liu *et al*. 1993), and the principle of virtual work (Wang and Gosselin 1998; Tsai 2000; Gallardo *et al*. 2003). Some researchers (Reboulet and Berthomieu 1991; Kim and Lee 1992; Kock and Schumacher 2000) concluded that the dynamics model needs to be simplified in order to be used in a real-time control system. Different methods can be applied depending on the situation and the requirements, i.e., the purpose of evaluating the dynamics, whether it is for control, evaluation or simulation purposes.

Control of the SP manipulator is still an open issue and the works reported are not rigorous. In the field of machine tools, the trend is to try to adopt existing hardware for controlling the PKMs. However, the use of existing hardware for controlling the manipulators will drastically penalize the performance of the system in the long term (Merlet 2002). Some researchers have suggested that each actuator can be controlled independently and robustly with a control law than a simple proportional-integral-derivative (PID) control system (Chiacchio *et al*. 1993). Another approach implemented an optimization scheme on top of a proportional-derivative (PD) control (Yurt *et al*. 2002). Wang (Wang *et al*. 1995) and Zheng (Zheng and Haynes 1993) presented a neural network control scheme and showed its superiority over kinematic control. A model reference adaptive control scheme has been proposed (Li *et al*. 2003) to control a machine tool, and the Popov hyperstable theory is utilized as the adaptive control law. Recently, a more advanced tracking control scheme has been proposed (Huang *et al*. 2004; Huang and Fu 2004) and feedback using a camera (visual-servoing) has been

implemented (Zuo *et al*. 2002; Dallej *et al*. 2006b; Andreff *et al*. 2007). Lastly, the combination of more than one single control strategies that takes advantage of multiple coordinated PKMs is another important field that is relatively unexplored.

2.6 Stewart Platform for Machining Applications

PKMs have several advantages over conventional industrial manipulators and machine tools. The main objective is to find applications where the PKMs can be best utilized according to their capability. These include applications where flexibility, accuracy and high loads are essential for success. Most of the previous research studies were focused on issues, such as kinematics, dynamics, singularities, workspace, etc. Relatively little effort has been focused on investigating the implementations of PKMs in the industry.

The trend in the manufacturing industry is towards shorter product life cycles and a larger variance of products. For example, the automotive industry is producing smaller batches and uses more common platforms and components. This increases the need for greater flexibility within the manufacturing systems and reconfigurable systems.

The cooperative manipulators proposed and developed in this research serves as a test-bed and a vehicle for exploring the characteristics of cooperative manipulators. The unique geometric structure of this cooperative manipulators is expected to confer a number of important performance advantages. Taken altogether, these advantages could revamp precision

manufacturing operations and stimulate new approaches to designing and machining parts, moulds and dies.

Currently, it is not known how well this proposed cooperative manipulators will perform as compared to conventional machine tools. Currently, the research is focused on investigating the attributes and limitations of this cooperative manipulators. After an initial characterization of this cooperative manipulators, the research focus is then expanded to techniques for enhancing the performance of the cooperative manipulators in machining, positioning and assembly applications. A main objective of this research is to develop the underlying measurement methods and technologies needed to achieve high levels of positioning accuracy and resolution. Micro actuators will be incorporated into the cooperative manipulators so that the strut lengths can be changed in precise micrometre-scale increments. Eventually, a system for self-calibration will be developed so that the cooperative manipulators can check its own performance and correct any detected inaccuracies.

Although PKMs are considered one of the most radical innovations since Computer Numerical Control (CNC) was found, a better understanding of the real advantages offered by PKMs versus conventional machining centres is still an on-going research issue. Some research efforts on the comparison of conventional machining centres with PKMs have not found any good standardization to compare them on an equal basis (Tlusty *et al*. 1999; 2000; Fassi and Wiens 2000; Neugebauer *et al*. 2000). Therefore, one would try not to develop a manipulator to compete with conventional machine

tools on accuracy and stiffness but rather on the flexibility of accomplishing three- and five-axis machining with respect to the development of a general PKM. The expected performances of the cooperative manipulators with respect to flexibility are:

1. Fewer number of setups are required,
2. Larger workspace volume, especially the range of orientation, and
3. Higher speed as a shorter processing time can be achieved with two platforms moving together.

CHAPTER 3

COMPUTER NUMERICALLY CONTROL MACHINE TOOL CONCEPTS

A CNC machine tool is positioned according to a pre-programmed path by means of special codes forming an NC program; an NC program consists of commands represented by letters, numbers and special symbols. These commands are used to manipulate the machine tool or work-piece to produce a required industrial part. Nowadays, almost all machining tasks are generated using computer-aided design (CAD) part design followed by computer-aided manufacturing (CAM) to process the solid model so as to obtain the tool paths required to cut the material. Modern CAD and CAM software are able to reflect changes in a part design almost instantly in the part program. This allows late changes to be included in the production cycle, but without interfering with the entire design process from the beginning.

The NC program controls the machine movements following a certain manufacturing technology and methodology. Thus, a CAD/CAM software package often comes together with standardized machine tools libraries that can generate certain set of instructions compatible to a particular selected type of CNC machine tool. Moreover, they develop speeds and feeds data automatically based on tool selection. The NC program is able to output the required shape and size of the tool, the speed and feed rate, and the orientation

of the tool relative to the work-piece. These parameters are different from one machine to another in terms of their definition and reference. The part program also prescribes a set of cutter location (CL) points assigned to cut the desired part. A CL point is a specific position at which an NC machine has to move a cutter to.

Before an NC program is produced, the CL data file is processed for a specific machine tool. This process is called post-processing. Thus, the post-processor of each machine to be used must be present in the CAD/CAM package. The post-processor has the ability to output the correct syntax for a particular machine tool and transform coordinate systems with respect to the specific arrangement within the machine tool. The standard syntax for most machine tools is known as G-Code and M-Code. The post-processor permits additional functions or modification to be added according to the machine controller, as well as variation in the machine capabilities. After the entire NC program has been generated, it is stored in a file which is then fed to the controller of the machine tool.

The fundamental function of the controller of a machine tool is to move the machine tool along a linear and/or circular path interpreted from the G-Code in the NC file. The controller may employ an interpolation technique to overcome the limitations of the machine tool drivers, which may cause tool chatter or breakage. Similarly, positional control is used in the control system of a SP such that many sampled points are used. The controller of the cooperative manipulators uses a standard industrial PID control scheme to move the actuators to certain positions with various speed settings. Fine

tuning is performed to obtain the optimum PID parameters to allow actuators to move at various speeds while maintaining good dynamic performance, such as faster settling time and lower overshoot. In a coordinative configuration, both Stewart platforms can move together in order to move the tool relative to the object with less travel time.

The orientation of the cutting tool or the work-piece in one or two rotary axes can be changed in addition to spatial motions along the Z, Y and X axes according to the values specified in the NC program. These rotations have axes known as a combination of C, B, and/or A axes. Since a SP is basically a six-legged structure which can perform 6-DOF motions, it has the potential to be used as a 5-axis or 6-axis machine tool.

3.1 Part Geometry Design

CAD/CAM systems are widely used in industries for designing products quickly and efficiently. This has resulted in seamless communication and automated tool path generation. These systems are able to model the parts, and perform measurements and interference checking with high precision. They are used as a virtual prototyping tool to evaluate design requirements and verify errors before the designed part is sent to be machined.

The output of a CAD system is a representation of a part in an electronic format. The tool path needs to be generated using either the CAD system or specialized CAM software. If CAM is used, a design must be transferred to the CAM system each time when there is a design change. This has the tendency of slowing down the production process. Currently, there is

no specific CAM system or post-processor for machine tools based on SP as SP has not been widely used for machining applications. Therefore, tool path generation is done using standard machine tool settings. Subsequently, the tool path has to be converted in order to run the tool path in a SP-based machine tool.

The procedure used to machine a part using a SP-based machine tool is described as follows:

1. Generation of the tool path in the work-piece coordinate system. This step usually requires a successive set of coordinates called the cutter contact (CC) points and the tool orientations in the work-piece coordinate system, which are distributed along a set of curves following a specified pattern, such as the zigzag or the spiral curves.
2. The CL data are computed from the set of CC points. The method for CC-CL data conversion must include the geometry of the cutting tool and the orientation of the tool relative to the work-piece. Various optimization techniques can be used at this step to generate optimal cutter location data.
3. Each CL point is transformed into the machine command. Converting coordinates of the CL points from the work-piece coordinate system to the machine coordinate system is called post-processing. The post-processor requires knowledge of the machine kinematics and the machine configuration. In SPs, this is also called inverse kinematics control.

4. The machine commands are converted into an NC part program in a format that can be interpreted by the machine controller. In this case, MatLab is used to achieve this.

The above sequence may or may not include optimization and verification steps. The optimization step involves defining a cost function to represent a certain type of error or a combination of several types of errors. The tool path is modified or reconstructed entirely in such a way that the cost function is minimized or at least decreased. The verification stage includes an actual machining or computer modelling of the material removal process which produces the output as a solid model. The solid model can be compared with the desired part and the efficiency of the optimization strategies can be evaluated.

In this research, the CL points are converted to NC codes in CAM and further processing of the tool path is done through analysing the NC codes. CL points data are actually more useful compared to NC codes as it is more generic and not specific to a particular machine tool. However, because of software limitation, the specific commercial CAM software used in this research cannot export CL points directly to MatLab. Thus, it is necessary to convert and process the NC codes.

3.2 PKM-based Machine Tool Advantages

A PKM-based machine tool has several potential applications. One example is its use in a flexible manufacturing environment. Since a PKM is generally a positioning device, it is possible to use an interchangeable end-

effector module to perform a variety of operations, such as drilling, milling, grinding, welding, and assembly. In addition, PKM machine tools can be designed to utilize high speed spindles to remove material quickly.

Conventional machine tools, which have a cantilever or gantry type of structure, have limited payload due to their inherently large moving masses of their components. This surfaces the needs for powerful actuator systems, more precise control systems, and larger components for stiffness requirement. On top of these, thermal shielding is necessary due to the large amount of heat from the motors and bearings. Designing a PKM machine tool could lead to lower cost (Weck and Staimer 2002).

A PKM machine tool can have greater dexterity because it minimizes the moving mass. In operation, PKM only moves its end-effector mass. The work-piece is clamped onto the end-effector and remains stationary. Higher acceleration and speeds are achievable. It also minimizes the error on the final position of the end-effector because relative errors in the actuator positions are averaged instead of added, which is the case in serial manipulators. PKM consists of fewer mechanical bodies and each structural linkage is built from the same components. It is suited for modular design that would reduce the service and maintenance costs. Finally, the operating cost is lower because of the lower energy consumption and faster material removal rate (Fassi and Wiens 2000).

CHAPTER 4

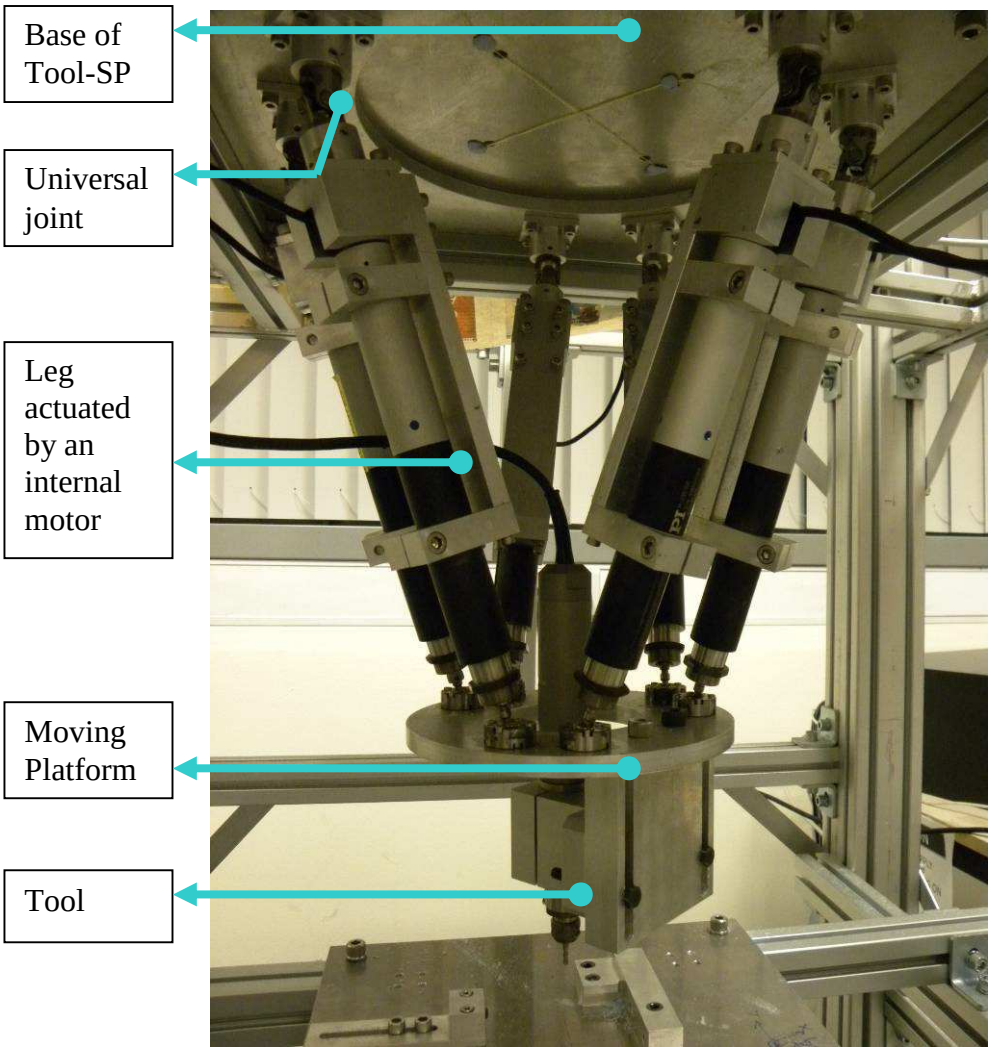
THE COOPERATIVE MANIPULATORS DESIGN AND IMPLEMENTATION

4.1 Cooperative Manipulators Structure Description

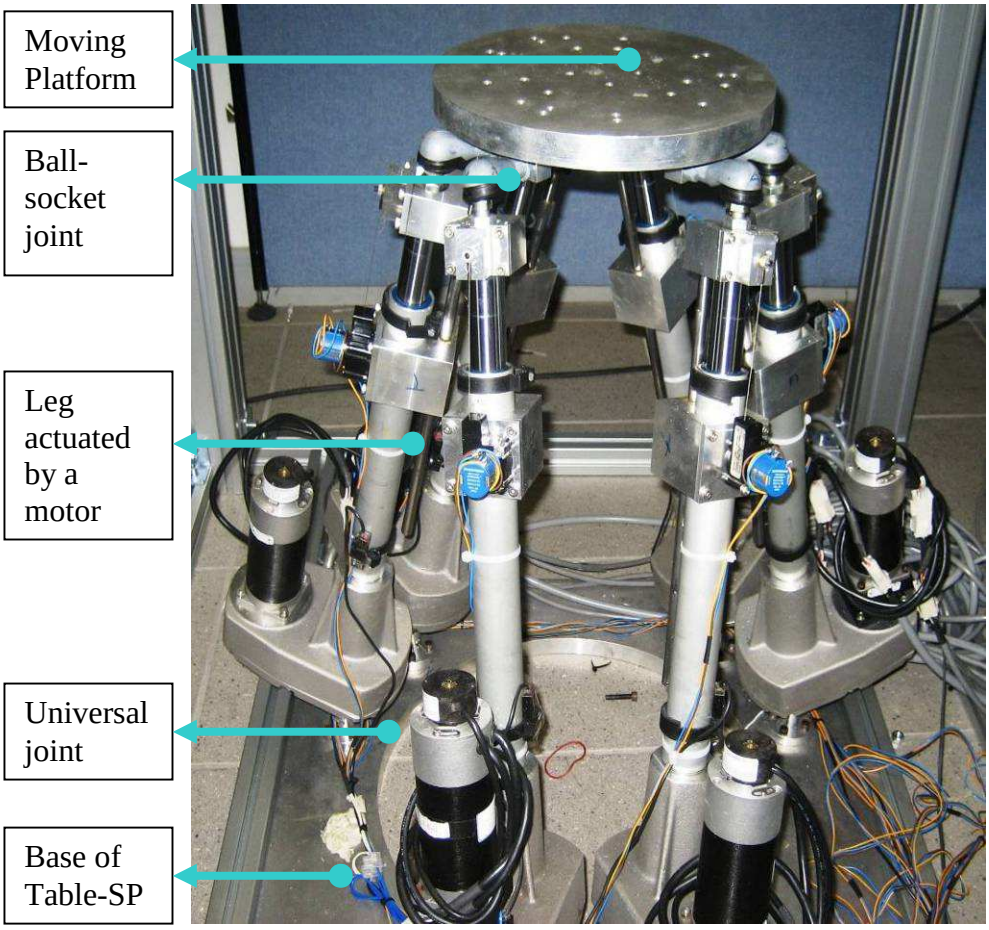
Two Stewart platforms, namely, a tool-SP (Figure 4.1a) and a table-SP (Figure 4.1b), have been built in this research to form cooperative manipulators to be used as a machine tool. Both Stewart platforms have similar structures. The spindle is attached to the tool-SP in Figure 4.1a. This tool-SP, mounted inversely onto a frame, is the main manipulator in this research, and it is referred to as the tool-SP. The six actuators installed between the moving platform and the base produce the motion. Universal joints are used to connect the actuators to the base and ball-socket joints are used to connect the actuators to the moving platform, which holds the spindle.

The second SP is installed on the frame below the tool-SP. This SP is named the table-SP. The table-SP works with the tool-SP, and it is thus referred to as the table-SP in this research. Figure 4.2 presents the configuration where the tool-SP and the table-SP are both installed on a frame. In this configuration, the work-piece is clamped onto the moving platform of the table-SP. The table-SP can move the work-piece relative to the tool-SP to provide access to locations further on the work-piece which are outside the workspace of the tool-SP. The table-SP has a larger dimension to provide

larger movement range for the work-piece to achieve a larger overall workspace. This configuration is named the cooperative PKM machine tool.



(a) Tool-SP (Inverted installation)



(b) Table-SP

Figure 4.1 Two Stewart platforms

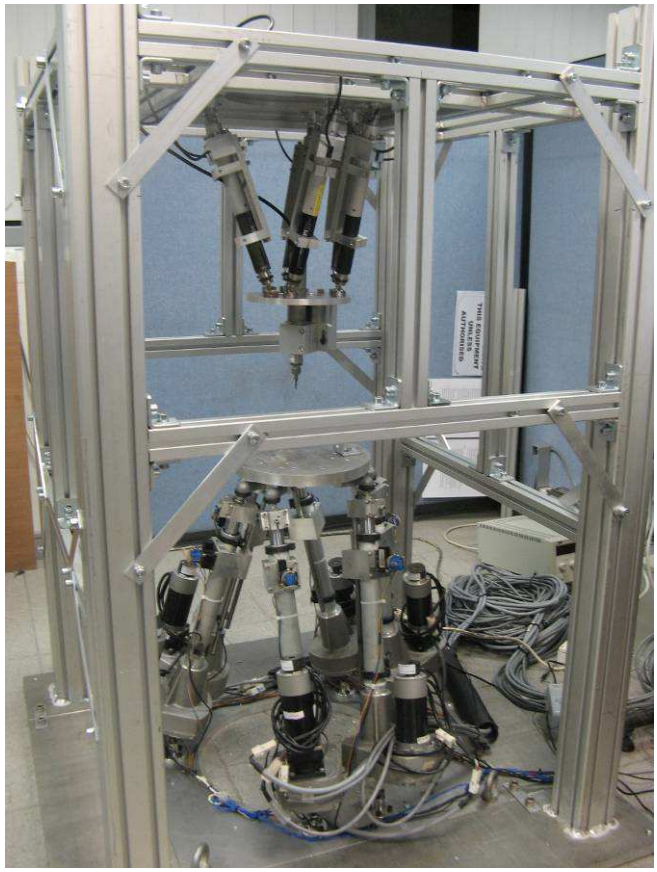


Figure 4.2 Extended Configuration of the cooperative manipulators

4.2 Coordinate Systems and Kinematics

Figure 4.3 shows a schematic of a Stewart platform with its kinematic designation. A ball-socket joint connects each prismatic actuator to the moving platform. Similarly, a universal joint connects each actuator to the base. This arrangement allows the moving platform to move based on the lengths of the prismatic actuators or the leg lengths.

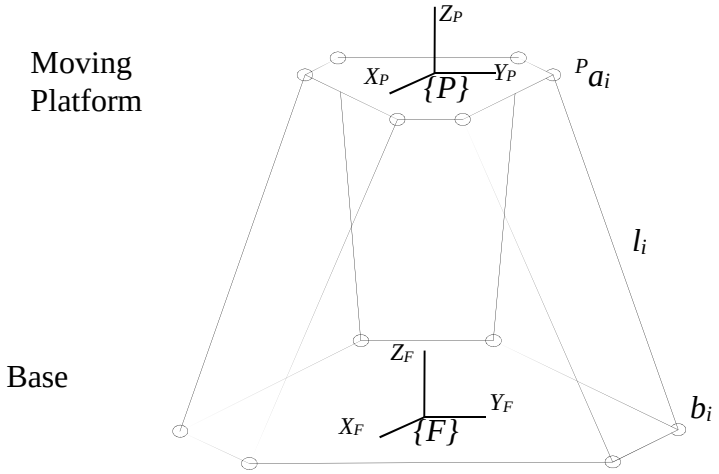


Figure 4.3 Schematic representation of the Stewart platform

There are two coordinate frames, namely, a fixed or base coordinate frame $\{F\}$ and a moving platform coordinate frame $\{P\}$. The position vector b_i denotes the position of the centre of the universal joint of leg i in $\{F\}$ and the position vector ${}^P a_i$ is defined in $\{P\}$ pointing to the location of the centre of the ball-socket joint of leg i. The legs are represented by vectors l_i defined in $\{F\}$. If ${}^F\boldsymbol{R}_P$ and $q = [q_x q_y q_z]^T$ are the rotation matrix and a position vector expressing the pose (orientation and position) of $\{P\}$ relative to $\{F\}$, l_i can be calculated using Equation 4.1.

$$l_i = {}^F\boldsymbol{R}_P \, {}^P a_i + q - b_i \tag{4.1}$$

This equation is well known as the inverse kinematic formula. The leg length can be computed by computing the length of this vector. The values of the leg lengths must take into consideration the initial length offsets of the legs L_{Oi}, and they are calculated using Equation 4.2.

$$\lambda_i = \|l_i\| - L_{Oi} = L_i - L_{Oi} \tag{4.2}$$

On the other hand, the forward kinematic model that is used to calculate the pose given the leg lengths is a difficult problem which is usually solved using numerical methods. The pose can be denoted by a generalized vector $\mathbf{X}=[q^T\theta^T]^T$, where $\theta=[\theta_x\theta_y\theta_z]^T$ describes a set of Euler angles of the orientation of $\{P\}$ relative to $\{F\}$. A Newton-Raphson scheme is developed in this research to compute a solution of the forward kinematic problem. This is an iterative method stated as follows.

1. Given an initial estimated pose $\mathbf{X}_g = [q_g{}^T\ \theta_g{}^T]^T$, calculate the inverse kinematic solution of this pose, which gives the six leg lengths, $L_g = [L_{g1}L_{g2}L_{g4}L_{g4}L_{g5}L_{g6}]^T$
2. Compute the partial derivative 6×6 matrix $\boldsymbol{J}$ of the pose vector, d$\mathbf{X}$, with respect to a small change of each leg displacement, where $J_{j,i} = dX_j / dL_{gi}$, i=1..6, j=1..6.
3. Update the current pose, $\mathbf{X}_g = \mathbf{X}_g + \boldsymbol{J}^{-1} * (L_a - L_g)$
4. Repeat steps (2)-(4) until $(L_a - L_g)$ is less than a certain pre-defined numerical threshold, and exit with the last $\mathbf{X}_g$ as the solution.

The initial pose can be set as the home position of the moving platform. The algorithm will give a robust solution of the correct pose if the pose is within the range of the manipulator movement and not near to any singularity configurations.

In this thesis, the subscript '2' is used to denote all the kinematic assignments of the table-SP. Therefore, Equations 4.1 and 4.2 can be rewritten for the table-SP as the follows.

$$l_{i2} = {}^{F2}\boldsymbol{R}_{P2}\,{}^{P2}a_{i2} + q_2 - b_{i2} \qquad (4.3)$$

$$\lambda_{i2} = \|l_{i2}\| - L_{Oi2} = L_{i2} - L_{Oi2} \tag{4.4}$$

4.3 Components and Control System

Both the tool-SP and table-SP are built from commercially available components. The links and connecting parts are fabricated in the workshop at the NUS Advanced Manufacturing Laboratory.

4.3.1 Tool Stewart Platform

Figure 4.4 shows the tool-SP before it was installed on the frame as a part of the cooperative manipulators. Figure 4.5 depicts the main components of the tool-SP. Figure 4.5a shows a Physik Instrumente M-235 series actuator which is a high resolution linear positioning actuator. It provides a motion of 50 mm with sub-micron resolution in a compact package. It has a preloaded ultra-low friction, heavy duty ball screw. This ball screw is driven by a closed-loop DC motor. Figure 4.5b shows the ball-socket joint. It is a HEPHAIST SEIKO ball-socket joint, SRJ006C. The movable parts of the ball-socket joint have a very low frictional resistance due to its rolling joint structure under preload, achieving high precision with little clearance. Compared with other ball-socket joints of similar characteristics, these joints have higher rigidity and they are also much smaller in size. They are optimal for achieving high precision, high rigidity and downsizing of the PKM. Figure 4.5c is a universal joint with needle bearing. It is designed to maintain low backlash for critical positioning applications required in robotics and instrumentation and is

excellent for continuous operation applications. The joint has rigid axial stiffness for push/pull loads.

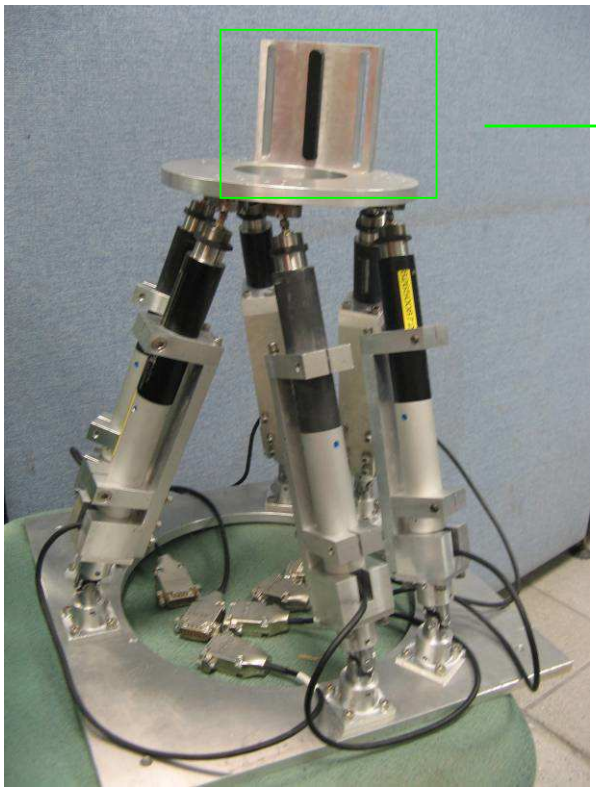
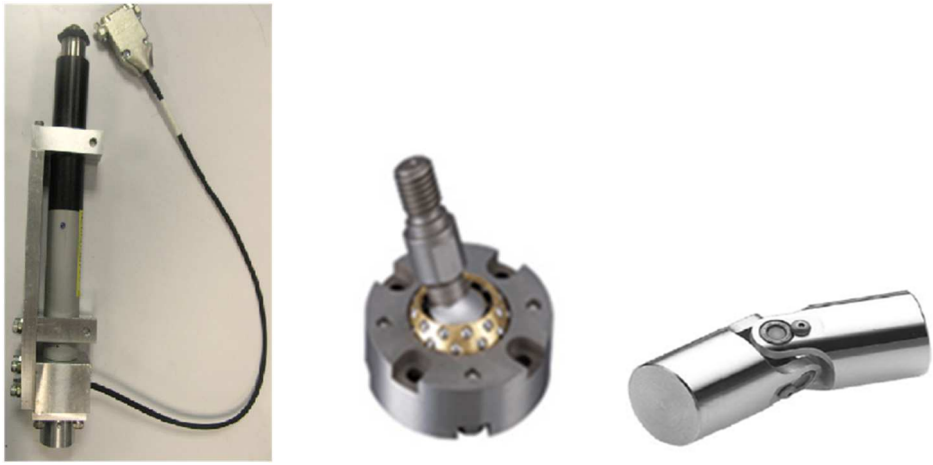


Figure 4.4 Tool-SP (not installed in the frame)



(a) (b) (c)

Figure 4.5 Actuator and passive joints in the tool-SP

The tool holder is attached to the moving platform. The tool holder is designed so that the tool height from the moving platform can be adjusted. There is an aluminium square block of thickness 90mmx20 mm, which is attached on top of the moving platform using two M10 screws (Figure 4.6). The spindle is fixed on to the aluminium block. The tool can be moved up and down through the slot on the aluminium block. Figure 4.6 presents this arrangement.

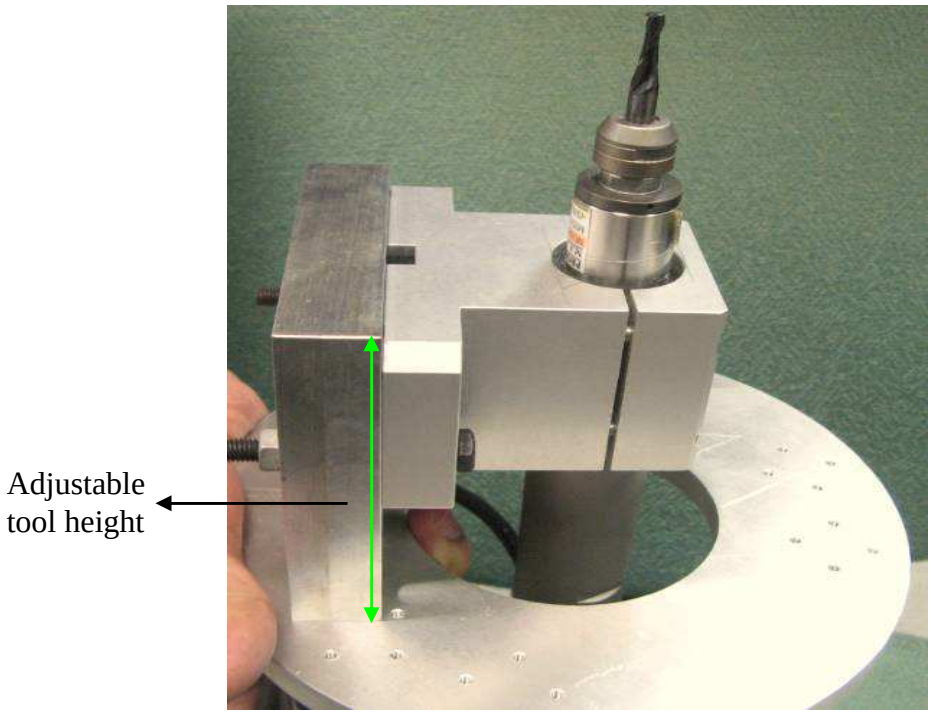


Figure 4.6 The tool attached to the moving platform

The controller for the tool-SP is the PC-based controller cards. One of the controllers is shown in Figure 4.7. It provides closed-loop digital servo-control and it comes with incremental encoder position feedback signals. The

axes can be programmed either independently or synchronously to allow multi-axis motion. Power is drawn from the PC power supply. Therefore, the power of the motor could be limited.

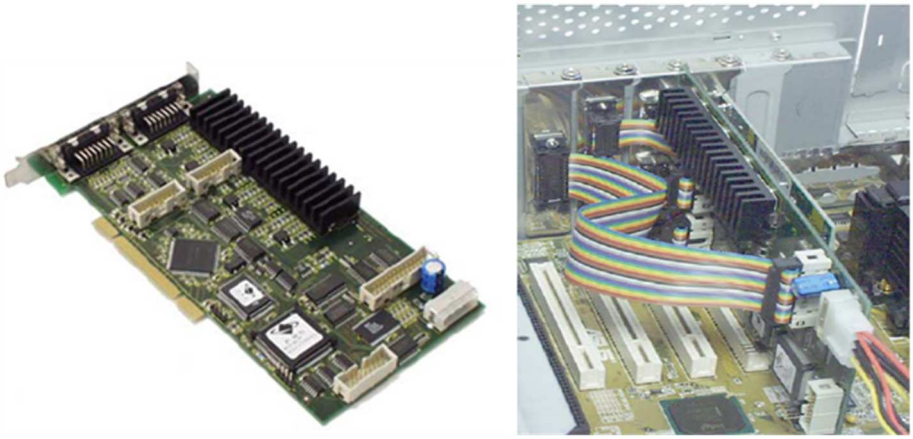


Figure 4.7 The PC controller card for the tool-SP

The PC controller card provides a wide range of functions for controlling the actuators. Each motor is controlled independently using a built-in PID controller, and a desired position and velocity can be specified. The controller card comes with C/C++ library that can be used for programming. The library can be called from a MatLab function to perform the operations.

4.3.2 Table Stewart Platform

The table-SP has a similar configuration as the tool-SP. It has larger actuators and passive joints. Its actuators are driven by servo motors with a dedicated power supply. Therefore, the power rating is higher than the tool-SP and the table-SP can move a larger payload. The actuators are also

controlled by a PC controller card. The main difference lies on the passive joint selection in the tool-SP and the table-SP. The table-SP is designed for a larger workspace and requires passive joints that have high maximum tilting angle. However, it is always a trade-off in the market between tilting angle and the joint clearance in the design of passive joints themselves with respect to price. The joint clearance affects the accuracy of the moving platform position and orientation, especially for the ball-socket joints. Higher joint clearance leads to higher errors in the Stewart platform kinematic chains. Thus, although the table-SP has a larger workspace, the expected accuracy is lower compared to the tool-SP.

Figure 4.8 shows the table-SP without the frame. The kinematic chains are the same as the tool-SP. The table-SP is designed for carrying the work-piece which can be fixed on top of its moving platform using fixture. The motor at each leg is driving the leg length through a pulley and a ball screw to convert the rotary motion to linear motion. Such a design allows the motor to be larger and hence has a higher power rating to drive heavier load. A high precision encoder is installed on the rotor of each motor to sense the rotary motion and provide feedback to the controller.

Figure 4.8 The table-SP (outside the frame)

The ball-socket joint and the universal joint, which connect the legs to the moving platform and the base, are shown in Figure 4.9. There are components that are fabricated in the workshop besides the joints. The passive joints are selected according to the size and power rating of the actuator. The ball-socket joints and the universal joints have working angle limits of 35 and 60 degrees, respectively.

Figure 4.9 The passive joints of the table-SP

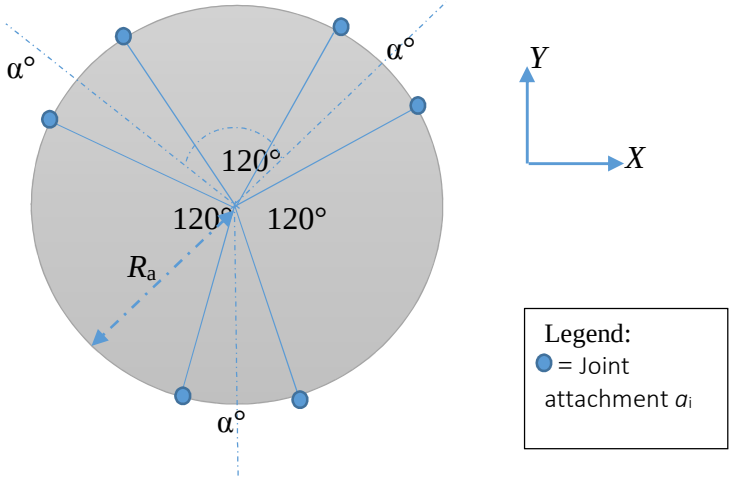
The table-SP legs are actuated by six servo motors controlled by a SPiiPlus PCI controller card installed on the PC. The controller can perform various functions and generate motion profile for testing each motor. The vendor has supplied the program manager by which the servo parameters can be adjusted. In addition, it provides 10 internal program buffers than can be activated simultaneously. It also has an extensive C/C++ library that can be integrated with MatLab.

4.3.3 Design Consideration

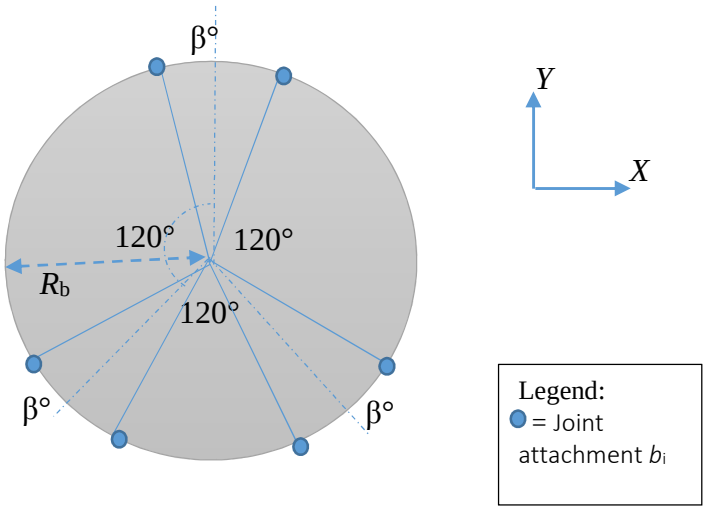
Machine design is a complex subject. In this section, the dimensional synthesis of the Stewart Platform is elaborated. Both SPs have undergone similar design processes. Design is carried out with emphasis on achieving maximum 3D workspace. The workspace of a Stewart platform is obtained using a discretization method (Merlet 2006) and the results with respect to

independent variables are plotted so that optimum geometric parameters can be determined. Mechanical kinematics constraints are also considered in the simulation so that the limitation and singularity of the movements of the platform can be predicted without driving the links to their limits. During the normal operation of the Stewart Platform, the singularity configuration should always be prevented to reduce the risk of uncertainty of the position and orientation as well as the possibility of overloading the joints of the platform that may cause breakdown.

Consider the tool SP which has been introduced in the previous section. There are four geometric design parameters to be considered. For simplicity, the main relationship between joints location attached to the base and to the moving platform is shown in Figure 4.10.



(a) Moving platform



(b) Base

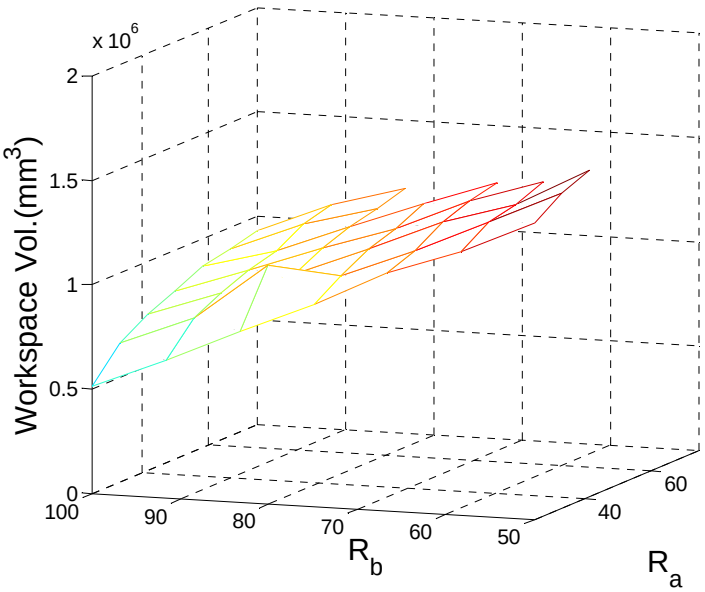
Figure 4.10 Relationship between joints positions

The base radius is always greater than the moving platform radius to ensure stability (i.e., $R_b \geq R_a$). Design optimization on dimensional synthesis can achieve certain performance, such as the workspace/footprint ratio. In real application, however, there are many factors that can affect the design decision, thus optimization depends on multi-criterion characteristics and other constraints. In this case, the workspace is computed with varying parameters of α, β, R_a, and R_b. The approach is to keep a pair of parameters constant (α, β) and (R_a, R_b) and plot the workspace characteristics with respect to two variables. The search range is defined for α, β, R_a, and R_b. It can be shown in Table 4.1.

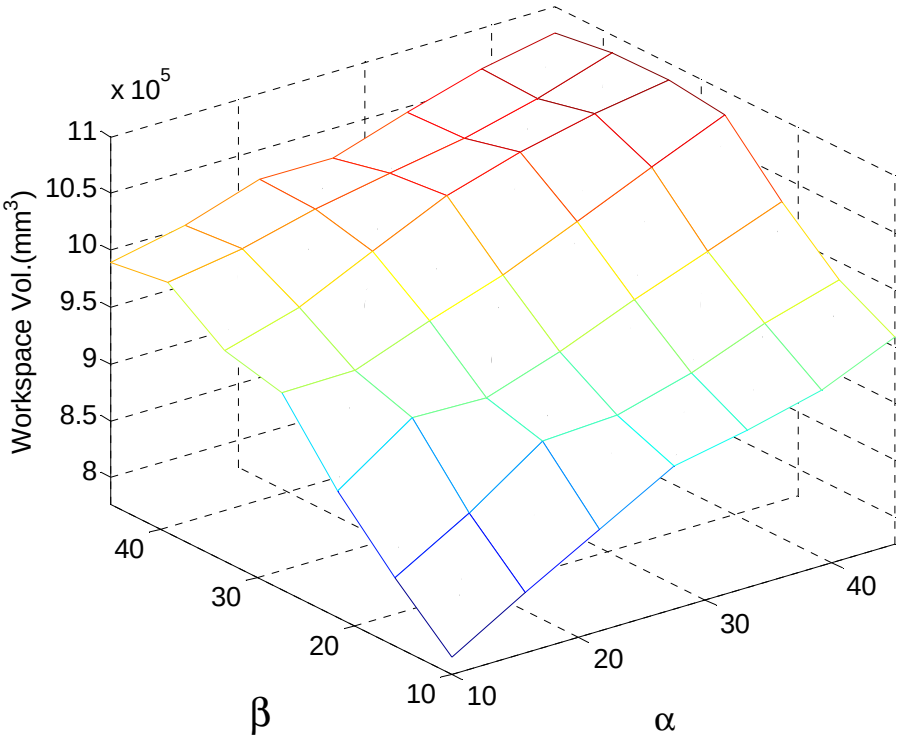
Table 4.1 Search range for the dimensional synthesis

	Minimum value	Maximum value
α and β	10°	45°
R_a	25mm	75mm
R_b	50mm	100mm

A three-dimensional surface plot is used to depict the resulting simulation. Figure 4.11 shows the 3D positional workspace volume of the SP with each pair of the dimensional design parameters held constant. From the results, a designer can then pick an optimum solution to the requirements, which can include workspace and footprint ratio. This process can be repeated for any combination of parameters of SP design.



(a) (α, β) is fixed to (30°,30°)



(b) (R_a,R_b) is fixed as (30mm,75mm)

Figure 4.11 Dimensional parameter synthesis of Stewart Platform

4.3.4 Joints Location

The six universal joints which connect the Stewart platform's legs to the base are located at point b_i, where i=1, 2,… 6. These points b_i are located along the perimeter of a circle and the centre of this circle coincides with the origin of the base frame $\{F\}$ at the centroid of the base platform. Similarly, the six ball-socket joints which connect the Stewart platform's legs to the moving platform are located at point a_i, where i=1, 2,… 6. The points b_i are

located along the perimeter of a circle which centre coincides with the origin of the moving platform frame $\{P\}$ at the centroid of the moving platform. This arrangement is shown in Figure 4.12 for the tool-SP. The circle radii r_F and r_P for the base platform and the moving platform are different for the tool-SP and the table-SP according to their dimensions. Furthermore, the joints are positioned based on angles θ_{ai} and θ_{bi} with reference from the positive X axis of the respective frame. Thus, the coordinates of the joint positions in the respective frames are determined as listed in Table 4.2.

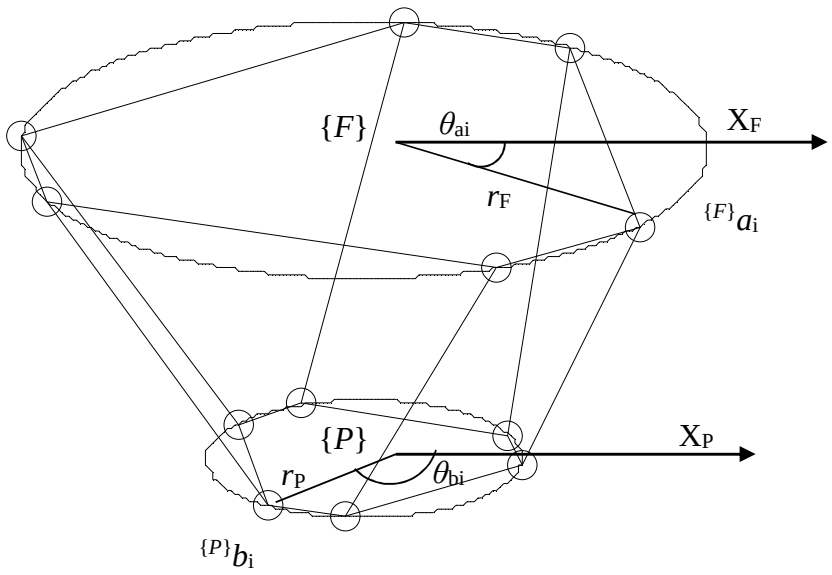


Figure 4.12 Joints location of the tool-SP

Table 4.2 Stewart platform joint locations (in mm)

	Tool Stewart platform		Table Stewart platform	
i	${}^{\{F\}}a_{\mathrm{i}}$	${}^{\{P\}}b_{\mathrm{i}}$	${}^{\{F\}}a_{\mathrm{i}}$	${}^{\{P\}}b_{\mathrm{i}}$
1	$\begin{bmatrix} 152.5\cos(180°) \\ 152.5\sin(180°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(150°) \\ 70\sin(150°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(45°) \\ 250\sin(45°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(15°) \\ 125\sin(15°) \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} 152.5\cos(210°) \\ 152.5\sin(210°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(240°) \\ 70\sin(240°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(315°) \\ 250\sin(315°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(345°) \\ 125\sin(345°) \\ 0 \end{bmatrix}$
3	$\begin{bmatrix} 152.5\cos(300°) \\ 152.5\sin(300°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(270°) \\ 70\sin(270°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(285°) \\ 250\sin(285°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(255°) \\ 125\sin(255°) \\ 0 \end{bmatrix}$
4	$\begin{bmatrix} 152.5\cos(330°) \\ 152.5\sin(330°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(0°) \\ 70\sin(0°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(195°) \\ 250\sin(195°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(225°) \\ 125\sin(225°) \\ 0 \end{bmatrix}$
5	$\begin{bmatrix} 152.5\cos(60°) \\ 152.5\sin(60°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(30°) \\ 70\sin(30°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(165°) \\ 250\sin(165°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(135°) \\ 125\sin(135°) \\ 0 \end{bmatrix}$
6	$\begin{bmatrix} 152.5\cos(90°) \\ 152.5\sin(90°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 70\cos(120°) \\ 70\sin(120°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 250\cos(75°) \\ 250\sin(75°) \\ 0 \end{bmatrix}$	$\begin{bmatrix} 125\cos(105°) \\ 125\sin(105°) \\ 0 \end{bmatrix}$

4.3.5 Frame Design

Before determining the dimensions of the frame structure, the kinematic model of the cooperative manipulators is built using MatLab. Based on mathematical analysis, the workspace can be determined (Figure 4.13a) which can be used to determine the size of the stationary frame. Based on the volume of the working space, the dimension of the frame structure is 1300 x 840 x 840mm (Height x width x length) (Figure 4.13a).

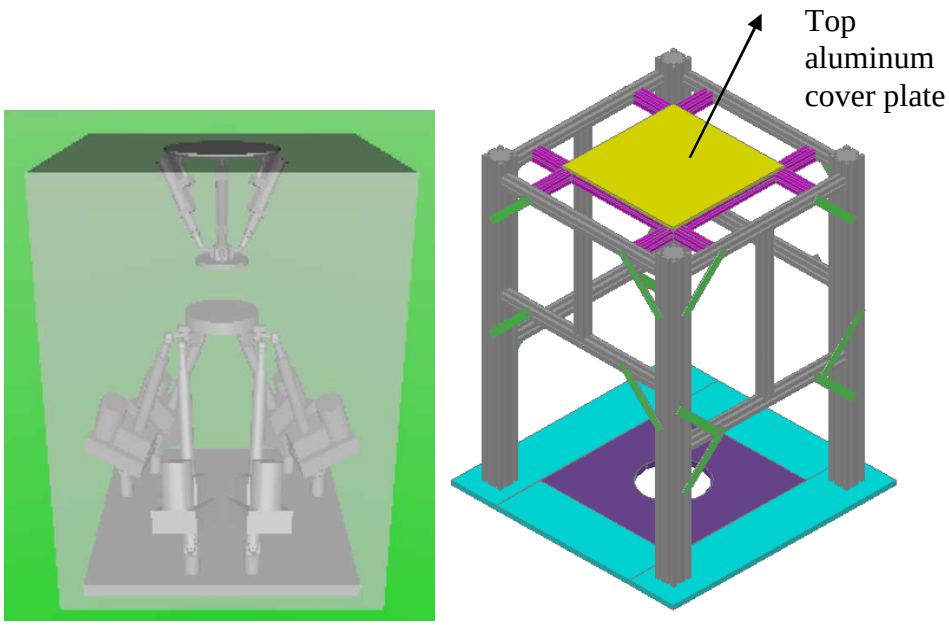
After determining the dimensions, the frame structure model was designed (Figure 4.13b). The main frame structure consists of aluminium bars of cross-section 40 x 40 mm and aluminium bars of cross-section 80 x 80 mm, which were used at the four corners to provide better rigidity and minimize vibration during the operating period.. In addition, four pieces of aluminium plates were assembled together to form a square base structure, surrounding the base plate of the table-SP.

For the top frame structure, all the aluminium bars (40 x 40 mm) were welded together and they can move up and down and can be tightened to the aluminium bars using angle brackets with M8 bolts and nuts. For the top aluminium cover plate, four M8 holes were drilled and tapped to hold the base plate of the tool-SP. Another twelve holes were also drilled to hold the cover plate onto the top of the frame structure.

4.4 Single Stewart Platform Configuration

In this configuration, the tool-SP is used alone. The work-piece is clamped onto an external fixed plate (referred to as the machining table),

below the tool-SP, which can be installed onto the frame as shown in Figure 4.15. Therefore, the table-SP is not used and it can be assumed that the table-SP is stationary holding the work-piece at one location. The home position of the tool-SP is not the zero position in the NC code programming. In the NC code programming, the coordinate system is selected with respect to the work-piece. Thus, a coordinate transformation is required. In Figure 4.14, a coordinate frame $\{W\}$ is attached to the work-piece. The work-piece is clamped onto the machining table relative to the coordinate frame $\{W\}$. By knowing the transformation between coordinate frames $\{W\}$ and $\{F\}$, the tool-SP can be controlled to perform the machining task according to the NC program.



(a) (b)

Figure 4.13 Frame Design for the cooperative manipulators

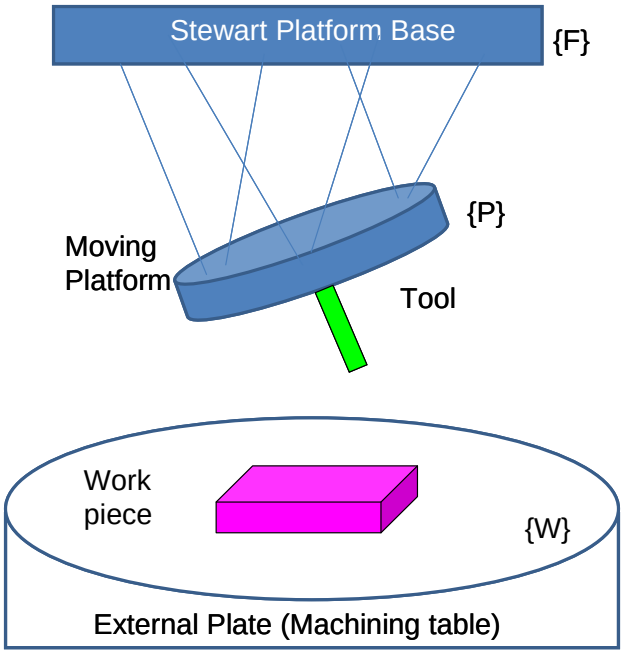


Figure 4.14 Single Tool Stewart platform Configuration

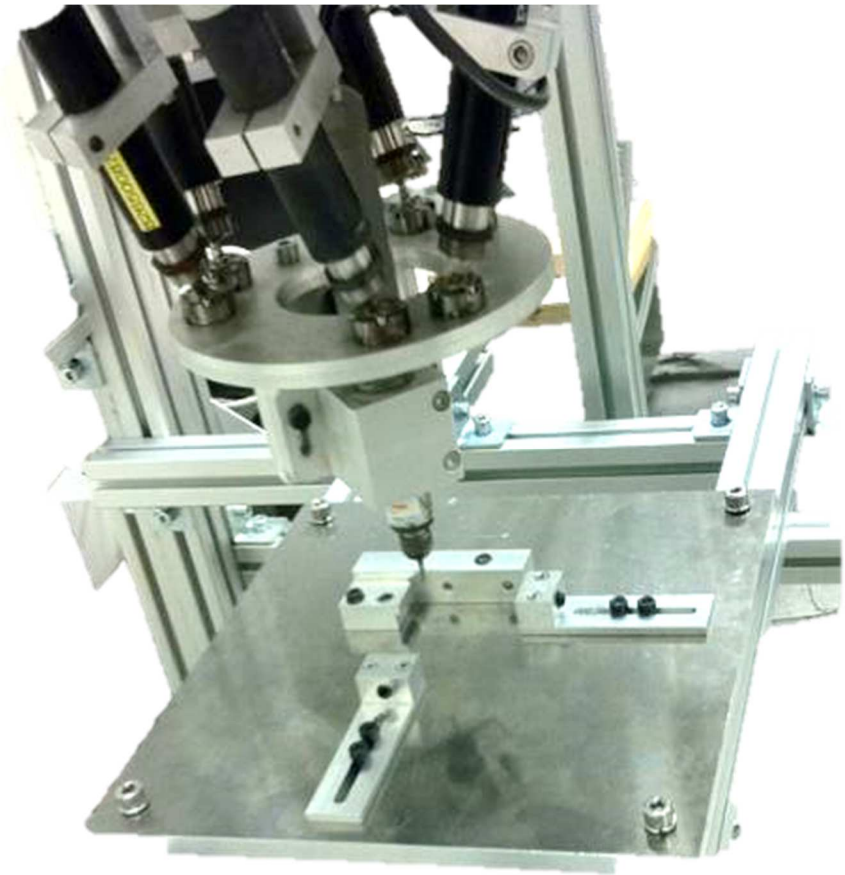


Figure 4.15 The machining table installed in the frame

4.5 Extended Configuration

In the extended configuration, the work-piece is clamped onto the moving platform of the table-SP, as shown in Figure 4.16. This table-SP acts as a movable table. The idea is to enable the work-piece to move beyond the workspace of the tool-SP so that the tool can access points that are not reachable in the single configuration. In this configuration, the bases of both the tool-SP and table-SP are attached with the coordinate frame {F}. The difference in this configuration is that the transformation of the work-piece coordinate frame relative to frame {F} is no longer constant. The work-piece coordinate frame is attached to the table-SP moving platform. However, because both SPs base are installed to the frame, the relation between coordinate frames of the tool-SP and the table-SP can be determined. Therefore, the machining task can be performed using both SPs in the extended configuration, which provides a bigger overall workspace than the single configuration.

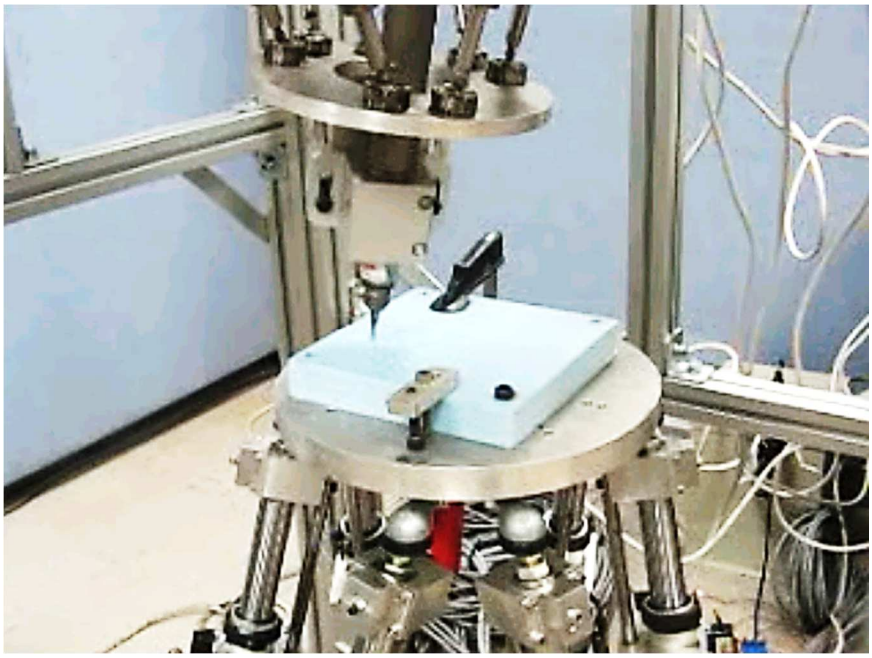


Figure 4.16 Machining a clamped work-piece on the table-SP with the tool-SP

CHAPTER 5

SIMULATION AND CONTROL OF STEWART PLATFORM

5.1 Workspace Analysis and Kinematic Constraints

It is important to develop algorithms by which the workspace of the Stewart platform can be determined and the effect of different designs on the workspace can be evaluated. The constraints that limit the workspace of a SP are modelled, namely, the ranges of the leg actuator, collisions between the legs, and the physical limitations of the passive joints (Harib and Srinivasan 2003). Additionally, singular configurations are to be avoided. In the workspace analysis, the following constraints are implemented.

1. Constraint on the leg actuator's length

   Each leg of the Stewart platform has a minimum and maximum length due to its mechanical limit. Therefore, each leg length must satisfy Equation 5.1, where $L_{i,min}$ and $L_{i,max}$ are the minimum and maximum lengths of the i^{th} leg, respectively.

   $$L_{i,min} < L_i < L_{i,max}, \; i = 1,2\ldots6 \tag{5.1}$$

2. Constraint on the range of the passive joints angle

   The ball-socket joints and the universal joints have their respective operating angles. Let θ_i and φ_i be the current angles of the ball-socket and universal joints, and they must satisfy Equation 5.2, where $\theta_{i,min}$

and $\varphi_{i,max}$ are the maximum operating angles of the i^{th} ball-socket joints and the i^{th} universal joints, respectively. The value of the current angle of the passive joints can be determined by the cosine rule to obtain the angle between two vectors.

$$\theta_i < \theta_{i,max} \; ; \; \varphi_i < \varphi_{i,max} \; , \; i = 1,2\ldots6 \qquad (5.2)$$

Let j_{ai} and j_{bi} be the unit vector along the axis of symmetry of the ball-socket joints and the universal joints respectively. Furthermore, let n_i be the unit vector along the leg axes, which can be computed by dividing the leg vector by its magnitude, $n_i = l_i / L_i$. The current angle of the ball-socket joints and the universal joints are calculated using Equation 5.3.

$$\cos^{-1} (j_{ai}^{T} \cdot n_i) \; , \; i = 1,2\ldots6;$$

$$\cos^{-1} (j_{bi}^{T} \cdot n_i) \; , \; i = 1,2\ldots6 \qquad (5.3)$$

3. Constraint on the collisions between the legs

   The legs can be modelled as a cylinder with a certain height and diameter, D_L. The centre of the cylinder is a line segment and coincides with the centre of the leg axis. Hence, in order to avoid any collision, the minimum distance between any two line segments corresponding to the legs of the Stewart Platform should be greater than or equal to D_L.

Since a 6-DOF workspace plot is not possible, only a 3D plot is used to represent the 3-DOF workspace graphically. The 3-DOF workspace can include any of three generalized coordinate variables of the moving platform

location. However, the most commonly used 3-DOF workspace is the 3-DOF positional workspace, which contains all the points reachable in a 3D Cartesian space with a fixed platform orientation.

The 3-DOF positional workspace of a Stewart platform is dependent on the orientation of its moving platform. The workspace can be defined as the 3D Cartesian space which is reachable by the centre of the moving platform where the origin $\{P\}$ is located. The algorithm to find the workspace is performed using a MatLab program, which algorithm can be summarized in Figure 5.1. This algorithm is known as the discretization method for the workspace computation of the Stewart platform. The 3D space is divided into finite number of discrete points representing a small region (a cell) of workspace with a certain size (a box with a length, a width, and a height). The box size can be approximated by choosing a size that contains the workspace. The algorithm is to check all the points whether they are reachable by the moving platform and store the result. After all the points have been tested, the results can be plotted.

The algorithm only checks the point of a corresponding cell. There is always a possibility that a portion of a cell is not reachable by the moving platform. Thus, the result of the discretization method is an approximation and it is more accurate than the actual volume with a smaller cell size. Let N_{reach} be the number of reachable points, the calculation of the volume of the workspace can be implemented by adding the volume of all the cells that are reachable using Equation (5.4), where ΔX x ΔY x ΔZ is the cell size in the Cartesian coordinates.

$$V = N_{reach} * \Delta X . \Delta Y . \Delta Z \ , \tag{5.4}$$

Figure 5.2 presents the plot of 3D positional workspace of the tool-SP with the orientation (0°, 0°, 0°). All the points inside the envelope in the figure are reachable. For this particular analysis, the cell size is ΔX x ΔY x ΔZ = 1mm x 1mm x 1mm, and the total volume is 386,680 mm^3. After obtaining the result of the workspace, the limitation of the position and orientation of the moving platform can be verified. Hence, the motion of the platform can be operated safely within the allowance of the workspace.

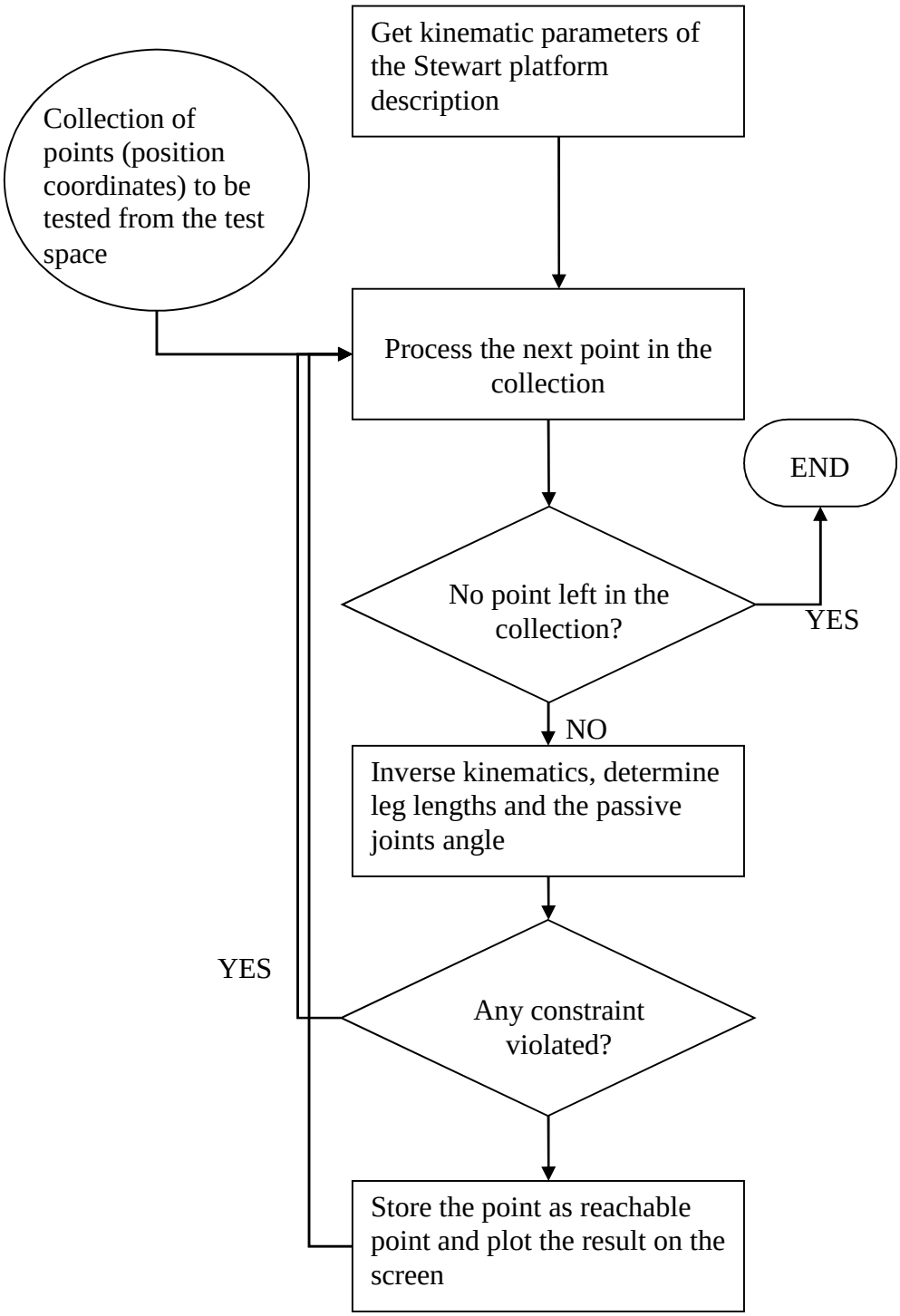


Figure 5.1 The flowchart to plot the workspace of the Stewart platform

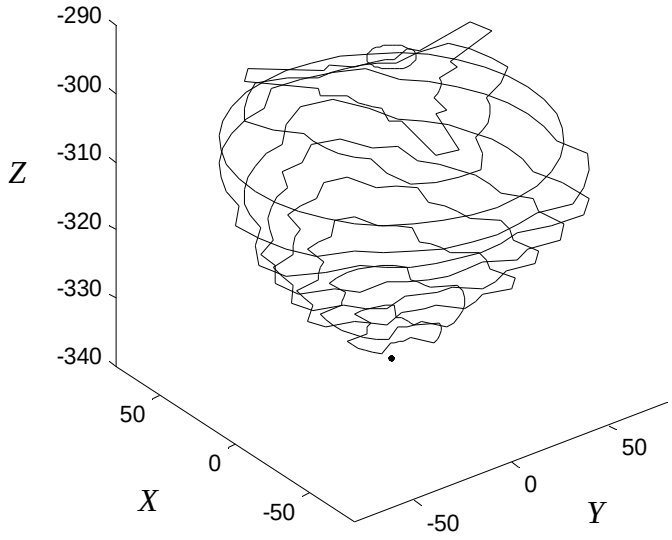


Figure 5.2 The tool Stewart platform workspace

5.2 Stewart Platform User Interface

The development of the Stewart platforms requires testing of the hardware and software. A user interface has been developed for controlling the Stewart platforms. The interface allows the user to define platform motion relative to the various systems: base coordinate frame, moving platform coordinate frame, and joint coordinates. In addition to these references, a joystick control has been added as a means to perform rapid movements of the Stewart platforms. The constraints of the passive joints that limit the movement of the moving platform are implemented after inverse kinematics calculation is performed. Therefore, the software can simulate the motion prior to its execution on a Stewart platform in real time.

The velocity and position input to the Stewart platform's controller can be provided in the following ways: manual input to the user interface,

pre-programmed data file, and 6-DOF joystick or a combination of joystick and keyboard. Figure 5.3 shows a user interface. In this interface, there are a few selections to be set in order to control the Stewart platforms, and they are listed as follows.

- Stewart platform to be controlled.
- Type of control: point-to-point motion, jog motion or spline motion, manual using joystick or keyboard, or read trajectory from a file.
- Sampling time and scale factor for the joystick control.
- The target location to move the moving platform for point-to-point control.
- Input file for the trajectory of the motion.
- Option to execute the prescribed motion in simulation or execution on the Stewart platform.
- Constraints that have to be verified to prevent damage.

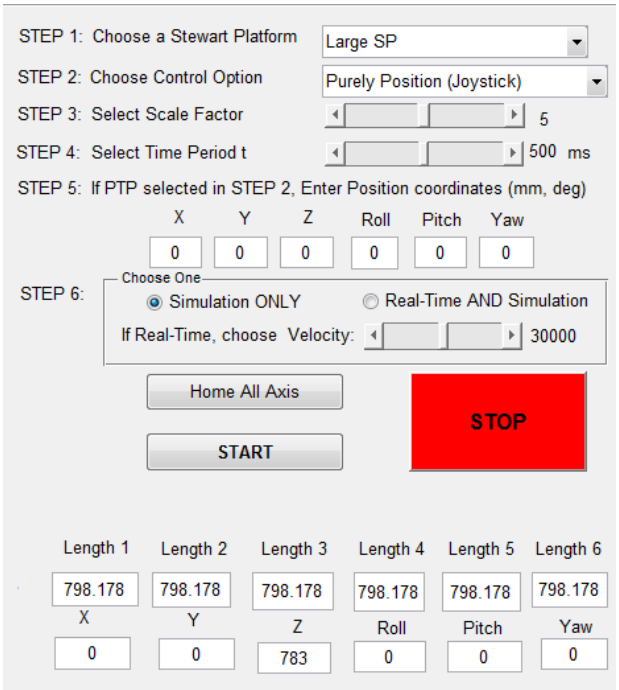


Figure 5.3 User Interface for Stewart platform control

The user interface is capable of guiding the user through various options of the interface in order to avoid any damage that may be done to the Stewart platforms. Most common errors can be avoided by checking the system status whenever possible. The interface displays various options from which the user can select using the PC pointing device (mouse). The user can opt to run the motion in graphical model shown in the interface without moving the actual Stewart platforms to check whether the execution of the motion is according to the desired trajectory.

When the program starts by establishing communication with the corresponding PC controller cards and homes all actuators to their zero reference positions. If the platform is initially high above the home position, performing homing directly is not advisable. This is because the homing function homes one axis at a time and hence if the platform is too far away

from the home position, the process can damage the joints. Therefore, there is an option to select the axis to perform the home function. The user is expected to know the state of the Stewart platform to select the axis to move.

Two additional options are implemented for joystick control, namely, scale factor and sampling time. The scale factor is the value of how much the joystick movement range is multiplied to determine the actual Stewart platform motion. The scale factor determines the range of the movement of the actual Stewart platform corresponding to each axis. This option enables the user to perform a rapid motion on the Stewart platform. Secondly, the sampling time is the waiting time between two subsequent data reads from the joystick. This is related to the response time of the Stewart platform to the joystick movement. The smaller the value, the more responsive is the Stewart platform and vice versa. There is a minimum sample time to prevent over speeding the moving platform. The joystick input can be treated as a distance relative to the current location of the moving platform or as a velocity. Both options can be selected from the user interface.

The graphical model of the Stewart platform moves when a move command is being executed. If the real motion option is selected, the Stewart platform also moves accordingly. The forward kinematics is implemented in the background such that for each movement, the values of the joint coordinates and the location of the moving platform are updated. Based on these values, the program is set to stop immediately and will halt all the Stewart platform motion as soon as one of the constraints is violated. Thus,

the state of the Stewart platform is always known to prevent violation of the constraints and joints over the limits during motion.

There is an option to specify the velocity of the motor of the Stewart platform legs when manual input using keyboard or automated motion is being selected. The input file should be specified if an automated motion is requested. This automated motion is especially useful for executing a pre-programmed trajectory, such as in NC machining. In summary, the user interface provides a basic and simple way to control the Stewart platforms.

5.3 Programming

The control of the user interface is implemented in MatLab which has built-in mathematical functions and matrix operations that make it ideal for inverse and forward kinematics calculation. In addition, it has a graphical user interface design environment (GUIDE) feature, which is useful for creating a user interface. In terms of control, MatLab can communicate with the controllers of the tool-SP and the table-SP simultaneously. This is supported by the MatLab capability to call C/C++ library, which is provided by the vendors.

As has been stated above, there are two input methods to the interface for controlling the Stewart platforms, namely, manual with keyboard or joystick, and automated by a pre-programmed data file. In addition, the controller can generate different types of motion profiles, namely, point-to-point motion, jog motion and spline motion. They are explained as follows,

1. Point-to-point motion: The response of the Stewart Platform is proportional to the joystick movement or the time period of a key pressed on the keyboard. Another option is to input the desired destination in the user interface or in the pre-programmed data file. This option uses a constant velocity for each leg. For machining, this motion scheme is used.
2. Spline motion: The joystick movement and keyboard input are interpreted as velocity in the corresponding direction of the moving platform. However, the values sent to the controller are the leg length values. Therefore, the program calculates the distance travelled per each sampling time period. The spline function is not suitable for manual input using keyboard or joystick because it needs to predict the next movement to determine the current velocity. It uses bi-cubic interpolation to calculate the required acceleration and velocity along the specified motion profile. Hence, this motion scheme results in lower position error.
3. Jog motion: The joystick movement and keyboard input are interpreted as velocity in the corresponding direction of the moving platform. The program sends the computed velocity values of each leg to the controller. The motion is smoother compared to the spline motion because there is no velocity interpolation. The side effect is that the resulting motion has lower accuracy. This motion is useful for rapid homing or retracting.
4. Interpolated motion: The previous motion types do not consider the kinematics of the Stewart platform. Since the motion control is done in

the joint coordinates, the resulting moving platform motion profile may not be as desirable in the specified trajectory because of the non-linear relationship between the location of the moving platform and the leg lengths. In order to control the moving platform trajectory correctly, the interpolated motion is developed. The interpolated motion uses many via-points between the starting and final position of the moving platform. Inverse kinematics calculation is performed in each via-point to update the required leg lengths. This process ensures that the moving platform follows the desired trajectory between the starting and final position.

The actuator leg lengths obtained after performing inverse kinematics are sent to the controllers after ensuring that the constraints are satisfied. Figure 5.4 summarizes the algorithm for the point-to-point motion in a flowchart. In each sampling, the variable in the selected coordinate system is decreased or increased according to the state of the input and the magnitude is set by the scale factor in the user interface.

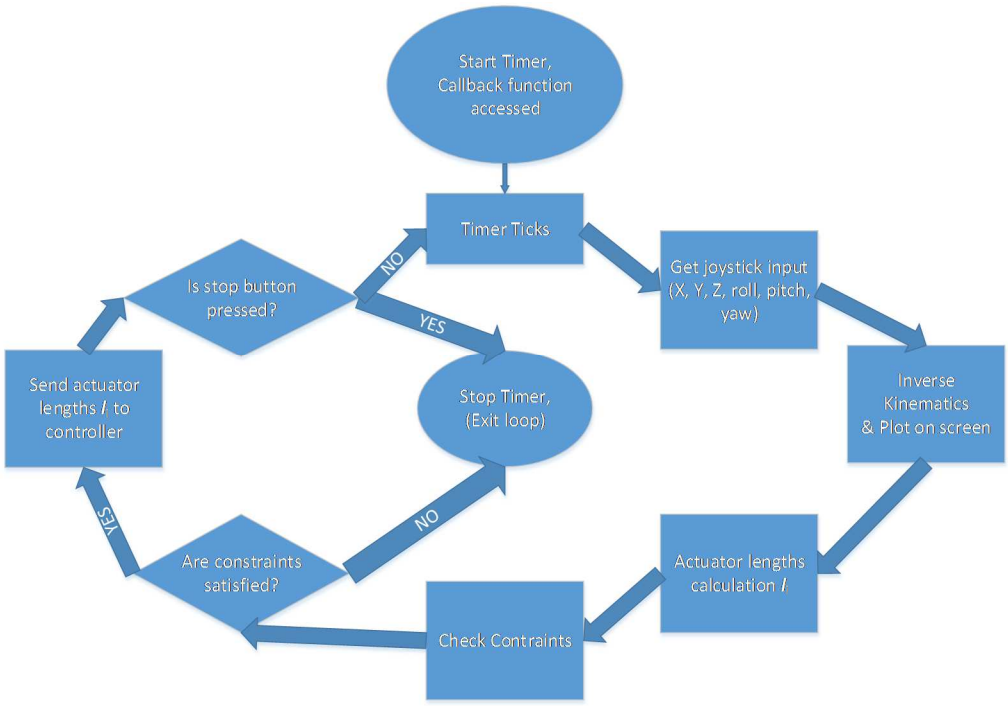


Figure 5.4 Point-to-point motion control flowchart

For the spline motion, the program has to calculate the leg lengths in each sampling period. The flowchart for the spline motion program is shown in Figure 5.5. The input is interpreted as velocity reference for the moving platform. In each sampling period, the new position is calculated. This is followed by inverse kinematics to determine the leg lengths. After a sequence of leg lengths has been acquired, it is sent to the controller.

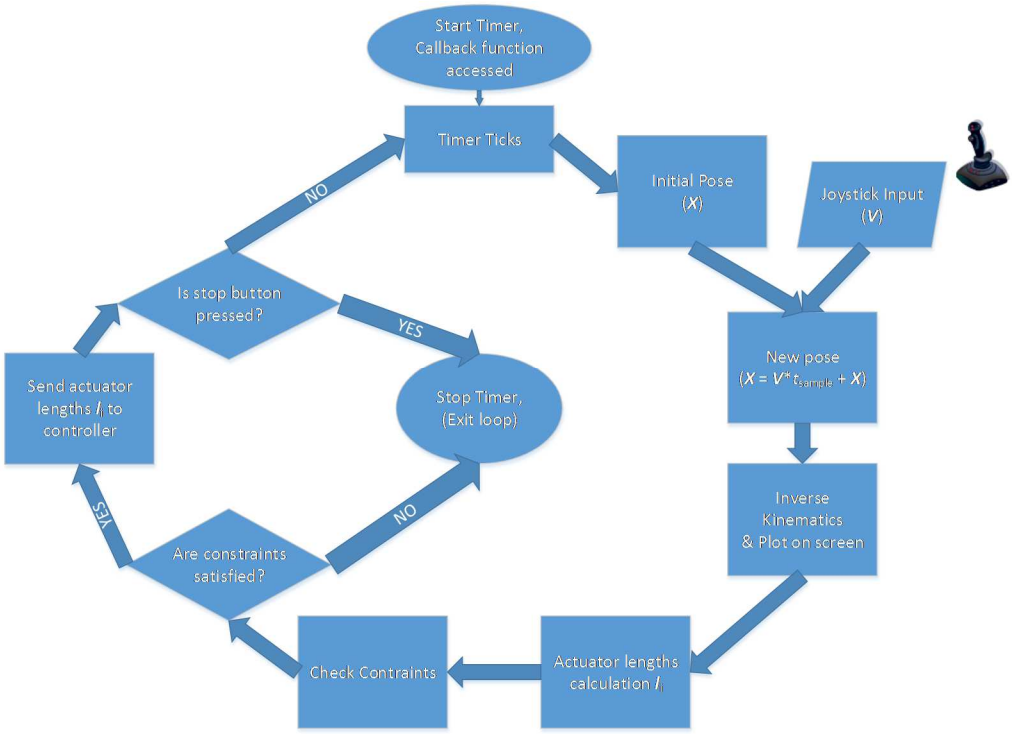


Figure 5.5 Spline motion control flowchart

The jog motion program interprets the input from the joystick or keyboard as a velocity reference of the moving platform. In each sampling period, the new leg lengths are calculated using inverse kinematics. The velocity of each leg is determined by dividing the difference between the new and the current lengths by the sampling time. The velocity is sent to the controller as soon as the calculation is completed. Figure 5.6 summarizes the algorithm flowchart for the jog motion program.

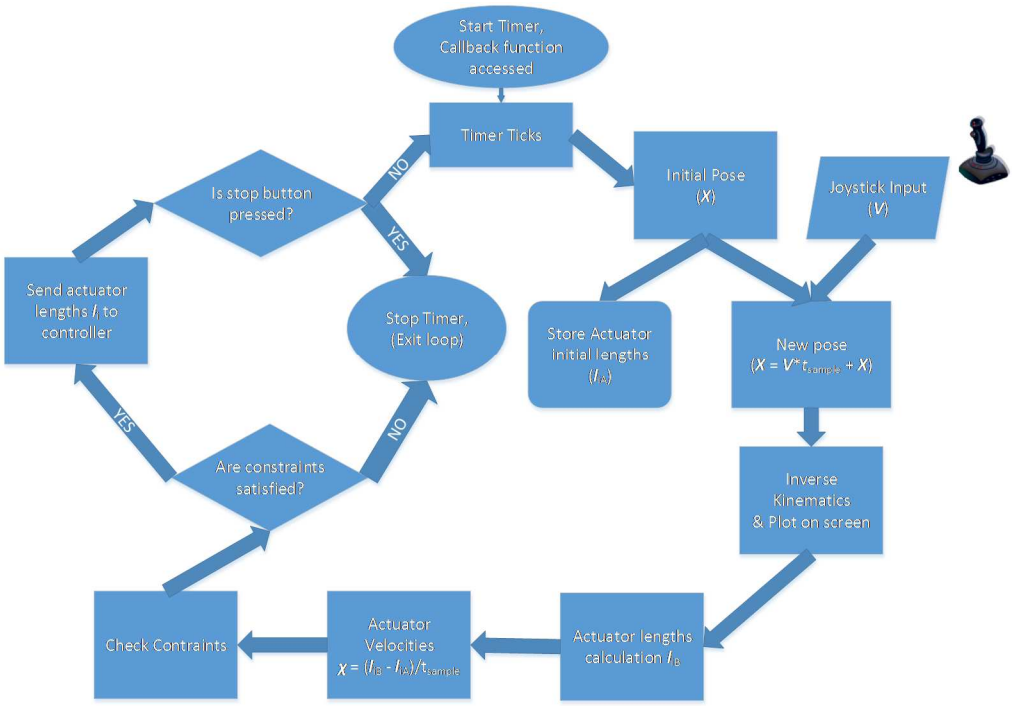


Figure 5.6 Jog motion control flowchart

Figure 5.7 shows the flowchart of the interpolated motion program. Interpolated motion is developed for executing the trajectory stored in a data file. The trajectory for machining can be obtained from the NC code for a certain job. In this program, more data points (via-points) between subsequent coordinates in the trajectory are added. By using many via-points, the discrepancies between the desired and the actual motion trajectory can be reduced significantly. As can be seen in Figure 5.6, the algorithm is a repeating point-to-point motion between the via-points. However, there is limitation on the minimum sampling period to allow sufficient time for the inverse kinematics calculation and communication with the controller while performing the motion with desired velocity.

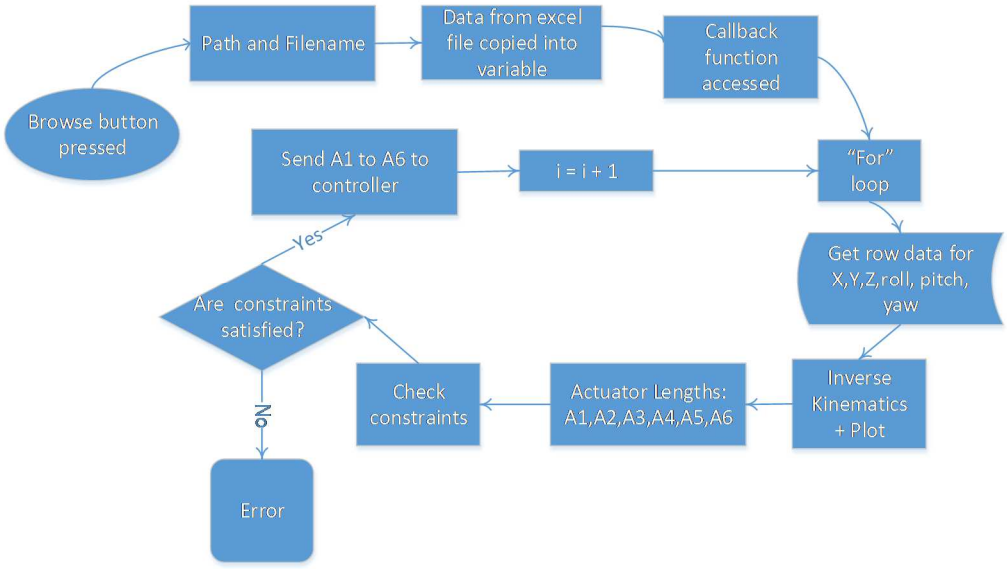


Figure 5.7 Interpolated motion control flowchart

5.4 Numerical Control Post-Processor for Stewart Platform

In this research, a commercial CAD/CAM software is used to design parts and generate the tool paths for these parts. The CAM output is the NC code to be sent to a standard CNC machine tool. The Stewart platform controller cannot interpret this NC code format. Hence, additional processing is required to convert the NC code to a data format which can be executed in the Stewart platform controller. This conversion is developed using MatLab.

The NC codes used for controlling a machine tool are based on certain conventions similar to computer programming languages. The NC codes form a sequence of codes called an NC program. An example of an NC program is shown in Figure 5.8. Table 5.1 summarizes a variety of NC codes and their functions. Each code in an NC program corresponds to a command to be executed by the CNC controller. The syntax of an NC code is a single character followed by a number. For example, to set a standard CNC machine

to accept input values of coordinates of the tool in millimetres, a specific G-code of ‘G21’ must be sent to initialize the machine.

```
%
O0000
G21
G0 G17 G40 G49 G80 G90
T2 M6
G0 G90 G54 X45.302 Y125.453 A0. S800 M3
G43 H2 Z10.
Z2.
G1 Z-10. F32.
G3 X42.013 Y139.208 R10.
G2 X63.987 Y175. R21.
```

Figure 5.8 Common NC program format

Table 5.1 NC codes and their functions

Character	**Function**
A	Additional rotary axis parallel and around the X axis
B	Additional rotary axis parallel and around the Y axis
C	Additional rotary axis parallel and around the Z axis
D	Tool radius offset number, Depth of cut for multiple repetitive cycles
E	User macro character, Precise designation of thread lead
F	Feed rate, Precise designation of thread lead
G	Preparatory function
H	Tool Length offset number
I	Incremental X coordinate of circle centre or parameter of fixed cycle
J	Incremental Y coordinate of circle centre
K	Incremental Z coordinate of circle centre or parameter of fixed cycle
L	Number of repetition
M	Miscellaneous function
N	Sequence or block number
O	Program number
P	Dwell Time, program number, and sequence number designation in subprogram Sequence number for multiple repetitive cycles
Q	Depth of cut, shift of canned cycles Sequence number for multiple repetitive cycles

R	Point R for canned cycles, as a reference return value Radius designation of a cycle arc
S	Spindle-speed function
T	Tool-function
U	Additional linear axis parallel to X axis
V	Additional linear axis parallel to Y axis
W	Additional linear axis parallel to Z axis
X	X coordinate
Y	Y coordinate
Z	Z coordinate

Since the codes beginning with ‘G’ and ‘M’ are most commonly used, these codes are usually referred to as ‘G-code’ and ‘M-code’. In the current work, the G-code and coordinate codes, such as ‘A’,’B’,’C’,’X’,’Y’, and ’Z’ are sufficient to control the Stewart platform as it does not have many preparatory or miscellaneous functions. A sequence of NC codes can be expressed as shown in the example shown next.

Word	Word	Word	Word	Word
N01	G00	X24	Y45	F40

Each word contains a character, followed by specific data (number) which can further be separated as shown below.

Char	Data	Char	Data	Char	Data	Char	Data	Char	Data
N	01	G	00	X	24	Y	45	F	40

By parsing and analysing these characters and data, the corresponding machine operation can be decoded and a tool path can be generated. A program block can consist of several G-codes. The list of G-codes that are used in the post-processor is shown in Table 5.2.

Table 5.2 G-codes used for Stewart platform and their meaning

Code	Group	Function
G00	01	Rapid traverse positioning
G01	01	Linear interpolation
G02	01	Circular and helical interpolation CW (clockwise)
G03	01	Circular and helical interpolation CCW (counter-clockwise)
G17	02	XY plane selection
G18	02	ZX plane selection
G19	02	YZ plane selection
G21	06	Input in millimetres
G90	03	Absolute programming command
G91	03	Incremental programming command

Based on the aforementioned explanation, a MatLab program is created to fetch a NC code file to parse each block and to translate it to coordinates in the Stewart platform coordinate system. These coordinates are stored in the data file and are used to control the Stewart platform according to the machining trajectory.

5.5 Stewart Platform Motion Emulation and Dynamics

The MatLab Virtual Reality Toolbox is used to model the complex constraints in the motion of a Stewart platform so that the entire geometric model (i.e., ball-socket joints, universal joints, platforms, compartments, and work-pieces) is considered. As a result, the process can be analysed in real-time. The collision avoidance algorithm uses the 3D model developed to compute the minimum distances between objects. The MatLab SimMechanics toolbox is used to model the dynamics of the Stewart platform. Figure 5.9 and Figure 5.10 show the representation of the 3D model and

dynamic control of the Stewart platform in the SimMechanics toolbox and Simulink, respectively. In this work, the integration of the motion animation with the Stewart platform control is implemented. The dynamics simulation is based on the data from the components vendor.

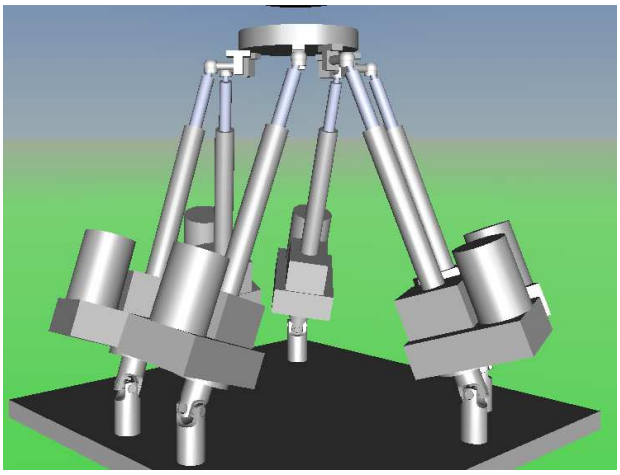


Figure 5.9 A 3D Stewart platform model

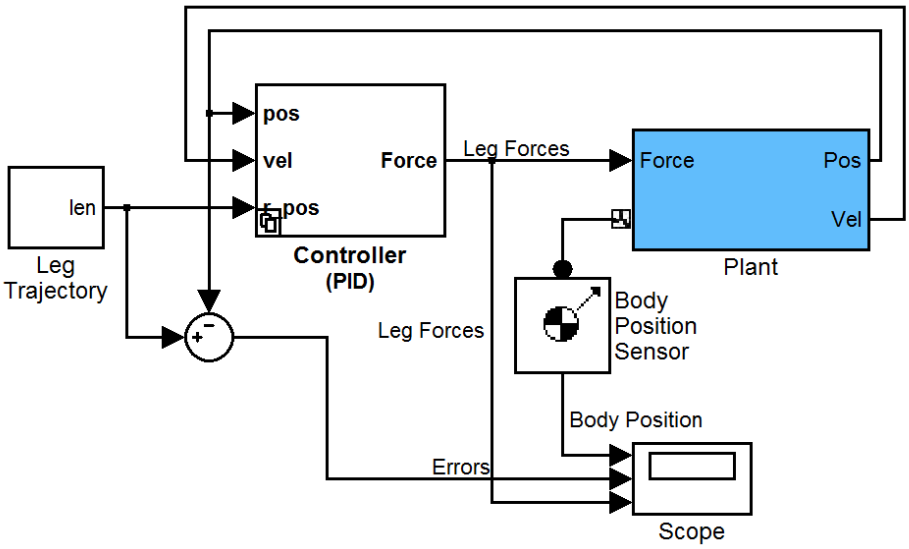


Figure 5.10 Stewart platform model in MatLab Simulink

CHAPTER 6

IMPLEMENTATION OF THE COOPERATIVE MANIPULATORS AS MACHINE TOOL

6.1 Coordinate Mapping of the Cooperative Manipulators

The task description of a spatial PKM generally consists of three linear coordinates and three angular coordinates describing the translation and rotation displacements of the end-effector relative to the object. The cooperative manipulators has a total of 12-DOF, consisting 6-DOF from the tool-SP and 6-DOF from the table-SP. For machining applications, the task description is defined in between three to five coordinates depending on the type of the machining job. For the 2.5- and 3-axis machining, there are three Cartesian coordinates. In complicated multi-axis machining, up to 2-DOF angular motion can be defined. These 1- or 2-DOF angular motions add two extra coordinates.

The developed post-processor is able to convert 5-axes machining to five corresponding coordinates in the Stewart platform coordinate frame. The machining can be performed using the tool-SP alone in the single configuration or with both the tool-SP and the table-SP in the extended configuration.

6.1.1 Single Stewart Platform Configuration

In the single configuration, the end-effector is the tool attached to the moving platform of the tool-SP and the object is the work-piece fixed onto the machining table. The detailed description of the single configuration with attached coordinate systems is presented in Figure 6.1. It is assumed that the base platform of the tool-SP and the work-piece are positioned in such a way that it is parallel to the machining table and the origin of frame $\{F\}$ and frame $\{W\}$ are aligned. The object or the work-piece is fixed onto the machining table below the tool-SP with a distance of h. There is a distance of d_t from the moving platform to the tool tip. The tool is attached to the tool-SP in such a way that the tool axis is coincident with the Z-axis of $\{P\}$.

The generalized coordinates used in the tool-SP are defined as the position and the orientation of the moving platform frame $\{P\}$ with respect to the base frame $\{F\}$.

$$^{F}\boldsymbol{X}_{P} = [q_x, q_y, q_z, \theta_1, \theta_2, \theta_3]^{T} \tag{6.1}$$

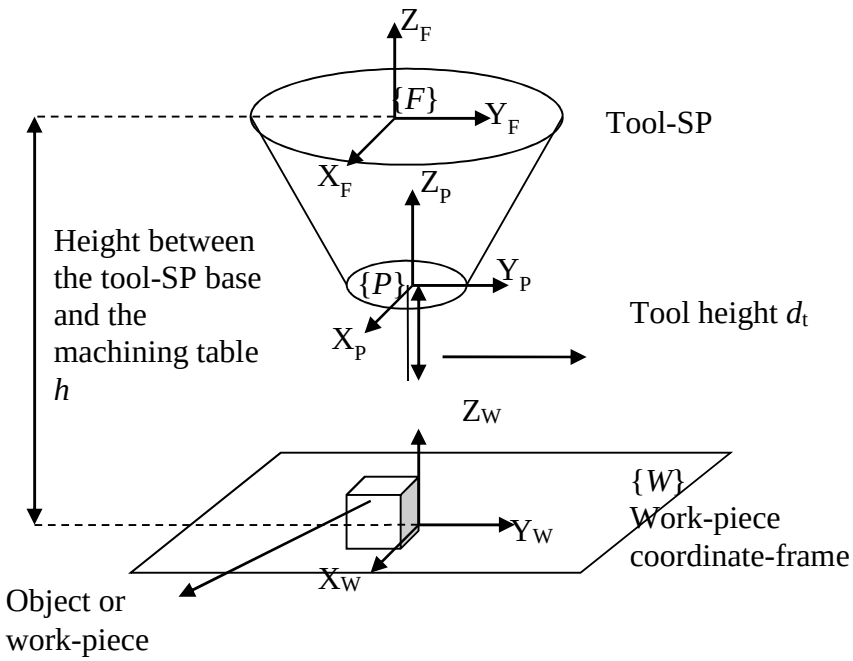


Figure 6.1 Coordinate system in the single configuration

The orientation coordinates must follow a certain convention for describing the rotation of the tool. In this research, the orientation of frame $\{P\}$ with respect to frame $\{F\}$ is specified by the following sequence of rotations. First, by rotating frame $\{P\}$ about axis Z_F with an angle θ_1, and rotating about the new axis X_P with an angle θ_2 , and then rotating about the new axis Z_P with an angle θ_3. Thus, the rotation matrix can be derived as Equation 6.2.

$${}^{F}\boldsymbol{R}_{P} = \boldsymbol{R}_{\theta 1}\,\boldsymbol{R}_{\theta 2}\,\boldsymbol{R}_{\theta 3}$$

$$= \begin{bmatrix} C_{\theta 1}C_{\theta 2}C_{\theta 3} - S_{\theta 1}S_{\theta 3} & -C_{\theta 1}C_{\theta 2}S_{\theta 3} - S_{\theta 1}C_{\theta 3} & C_{\theta 1}S_{\theta 2} \\ S_{\theta 1}C_{\theta 2}C_{\theta 3} + C_{\theta 1}S_{\theta 3} & -S_{\theta 1}C_{\theta 2}S_{\theta 3} + C_{\theta 1}C_{\theta 3} & S_{\theta 1}S_{\theta 2} \\ -S_{\theta 2}C_{\theta 3} & S_{\theta 2}S_{\theta 3} & C_{\theta 2} \end{bmatrix} \quad (6.2)$$

where $C_\theta = \cos(\theta)$ and $S_\theta = \sin(\theta)$, and

$$\boldsymbol{R}_{\theta 1} = \begin{bmatrix} C_{\theta 1} & -S_{\theta 1} & 0 \\ S_{\theta 1} & C_{\theta 1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_{\theta 2} = \begin{bmatrix} C_{\theta 2} & 0 & S_{\theta 2} \\ 0 & 1 & 0 \\ -S_{\theta 2} & 0 & C_{\theta 2} \end{bmatrix}$$

$$\boldsymbol{R}_{\theta 3} = \begin{bmatrix} C_{\theta 3} & -S_{\theta 3} & 0 \\ S_{\theta 3} & C_{\theta 3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

In general, the task description of a manipulator is input coordinates consisting of three linear coordinates and three angular coordinates describing the translation and rotation displacements of the end-effector relative to the object. For the PKM based machine tool, the input coordinates can be defined as the tool contact point with the work-piece. The vector representing the input coordinate in the coordinate frame $\{W\}$ is defined as Equation 6.4.

$${}^{\{W\}}\boldsymbol{X}_{\mathrm{i}} = [x\ ,\ y\ ,\ z\ ,\ \Phi,\ \Gamma\ ,\ \Omega]^{\mathrm{T}} \tag{6.4}$$

The coordinates x, y, z are the translation and the coordinates Φ, Γ , Ω are the rotation. The coordinate description in Equation 6.1 can be mapped to the 6-DOF coordinate system for the tool-SP. However, according the complexity of the machining task, fewer than 6-DOF may be required. For 3-axis machining, only the translation coordinates are provided by the CAD/CAM software as the tool orientation is not needed. For 4- or 5-axis machining, one or two orientation coordinate(s) is used.

For machining, the coordinates Φ, Γ are used for defining the 4th and the 5th rotation axes of the tool. The rotation coordinates Φ, Γ are defined as follows. First, by rotating about the new axis X_W with an angle Φ, and then rotating about the new axis $Z_W^{(1)}$ with an angle Γ. The last orientation coordinate Ω corresponds to the rotation of the tool along the tool axis which is redundant with the spindle rotation (cutter rotation for removing material). Hence, this last orientation is not needed for machining and it represents an extra DOF that could be used to optimize a certain index. This extra DOF comes from the fact that the machining task only requires maximum five-axis motion and the tool Stewart platform has six DOF.

The rotation angle coordinates can be mapped according to the description above. The first two input angle coordinates are mapped to the first two rotation coordinated of the tool SP, $\theta_1 = \Phi$, $\theta_2 = \Gamma$. The free DOF which is the coordinate Ω is mapped to the third rotation coordinates of the tool-SP, $\theta_3 = \Omega$. It can be set to a constant zero, $\Omega = 0$ for a trivial choice. The rotation matrix becomes Equation 6.5.

$$ {}^{F}\boldsymbol{R}_{P} = \begin{bmatrix} C_\Phi C_\Gamma & -S_\Phi & C_\Phi S_\Gamma \\ S_\Phi C_\Gamma & C_\Phi & S_\Phi S_\Gamma \\ -S_\Gamma & 0 & C_\Gamma \end{bmatrix} \tag{6.5} $$

Alternatively, it can be set such that $\Omega = -\Phi$ while maintaining the optimum pose of the moving platform (Bonev *et al*. 2002). The rotation matrix for this choice is given as Equation 6.6.

$$
{}^{F}\boldsymbol{R}_{P} = \begin{bmatrix} C^{2}{}_{\Phi}C_{\Gamma} + S^{2}{}_{\Phi} & S_{\Phi}C_{\Phi}\left(C_{\Gamma}-1\right) & C_{\Phi}S_{\Gamma} \\ S_{\Phi}C_{\Phi}\left(C_{\Gamma}-1\right) & S^{2}{}_{\Phi}C_{\Gamma} + C^{2}{}_{\Phi} & S_{\Phi}S_{\Gamma} \\ -S_{\Gamma}C_{\Phi} & -S_{\Gamma}S_{\Phi} & C_{\Gamma} \end{bmatrix}
$$

(6.6)

The translation coordinates [*x*,*y*,*z*] describe the position of the tool contact point with respect to frame {*W*}. These coordinates can be mapped to the position of the tool contact point ${}^{F}T_{c}$ (*x*',*y*',*z*') by transforming the input coordinates to the tool-SP base frame {*F*}. The transformation of frame {*W*} with respect to frame {*F*} is given as Equation 6.7 based on the assumption stated above.

$$
{}^{F}\boldsymbol{T}_{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.7}
$$

Thus, the *x* and *y* coordinates can be directly equated with the coordinates of tool-SP moving platform, *x*' and *y*'. The *z* coordinate is translated to *z*' with offsetting the values according the location of the origin of frame {*W*} with respect to frame {*F*}, which is the height *h*.

$$
x' = x \; ; \; y' = y \; ; \; z' = z - h \tag{6.8}
$$

Furthermore, since there is a distance between the tool contact point and the centroid of the moving platform, the position of the moving platform must be adjusted to achieve the correct location of the tool. This issue can be

described in Figure 6.2. The vector d_t . $\mathbf{v}_t$ is parallel to the tool axis as shown in Figure 6.2. Therefore, the moving platform (the origin of frame $\{P\}$) position with respect to frame $\{F\}$ in order to achieve the tool contact point at ${}^{F}T_c$ can be calculated as Equation 6.9. where $\boldsymbol{v}_t$ can be obtained as the unit vector of the Z axis of the moving platform with an opposite direction. The coordinate mapping can be summarized in Figure 6.3.

$${}^{F}O_P = {}^{F}T_c - d_t \,.\, \boldsymbol{v}_t \tag{6.9}$$

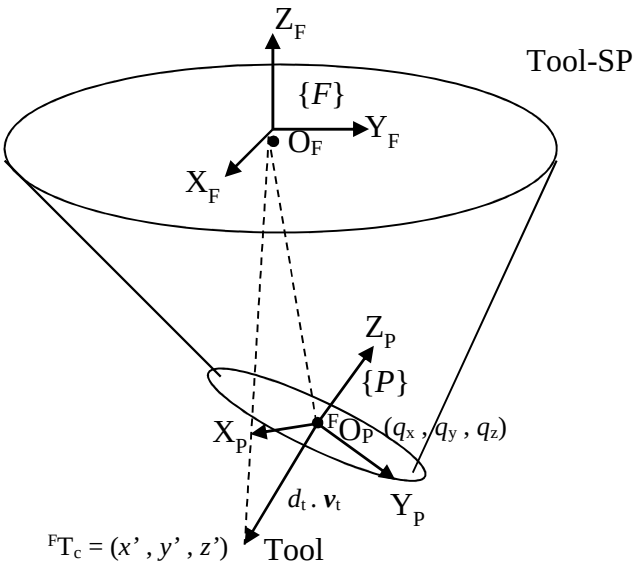


Figure 6.2 The moving platform position to reach the tool contact point

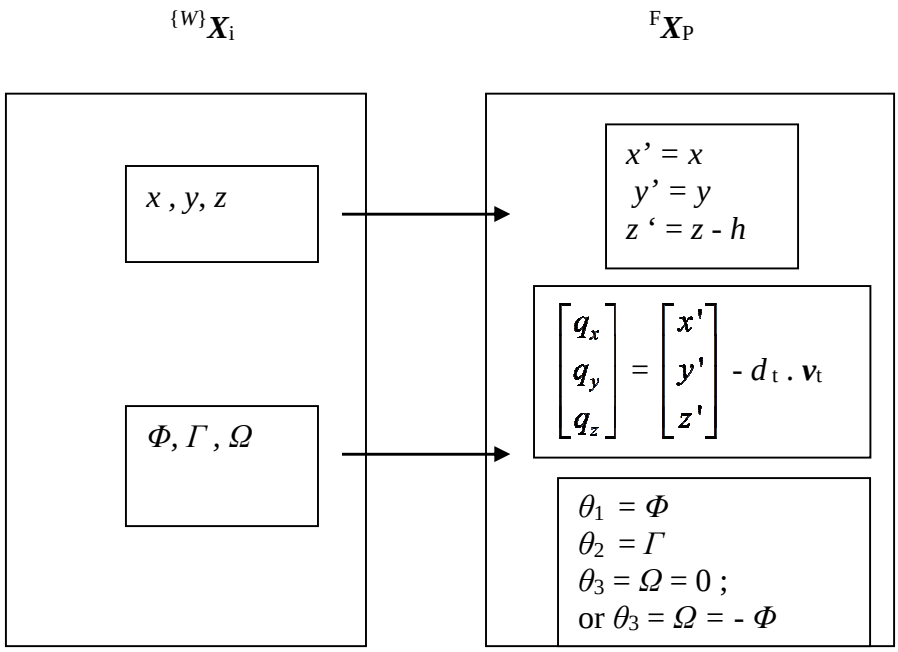


Figure 6.3 Coordinate mapping in single configuration

6.1.2 Extended Configuration

In the extended configuration, both the tool-SP and the table-SP are utilized. The coordinate system is presented in Figure 6.4. The table-SP is placed below the tool-SP and installed in the frame. The height difference between the Stewart platform bases is *h2*. The base frames {*F*} and {*F2* } are made parallel to each other and their base frame origins are aligned. Each SP has six DOF, hence the cooperative manipulators in this extended configuration has a total of 12 DOF. The generalized coordinates used in the table-SP are defined as the position and the orientation of the moving platform frame {*P2*} with respect to the base frame {*F2*}.

$$^{F2}\boldsymbol{X}_{P2} = [q_{x2}\ , q_{y2}\ , q_{z2}\ , \theta_{12}\ \theta_{22}\ , \theta_{32}\ ]^{T} \tag{6.10}$$

The transformation matrix of frame $\{F2\}$ with respect to frame $\{F\}$ is given as Equation 6.11.

$$ {}^{F}\boldsymbol{T}_{F2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.11} $$

The coordinate mapping is formulated to position and orientate both the tool and the work-piece in order for the tool to reach the desired tool contact point on the work-piece by using both the tool-SP and the table-SP. This means in order to fulfil the coordinate map, the tool-SP must move its moving platform in such a way that the tool tip can reach the desired tool contact point. This tool contact point is determined from the input coordinate ${}^{\{W\}}\boldsymbol{X}_{i} = [x, y, z, \Phi, \Gamma, \Omega]^{T}$.

The work-piece in the extended configuration is fixed onto the moving platform of the table-SP. It is assumed that the work-piece is setup such that the description of the input coordinates in frame $\{W\}$ is the same as the input coordinates in frame $\{P2\}$. Thus, one can also write the input coordinate as Equation 6.12.

$$ {}^{\{P2\}}\boldsymbol{X}_{i} = [x, y, z, \Phi, \Gamma, \Omega]^{T} \tag{6.12} $$

In addition, the tool-SP carries the tool so that the tool tip can reach the coordinate described by ${}^{F}T_{c}$, which is the input coordinate relative to frame $\{F\}$ (the base frame of the tool-SP). Figure 6.5 explains the configuration of the attached coordinate systems. The tool contact point on

the work-piece and the tool contact point that must be reached by the tool tip are the same point such that, $\boldsymbol{X}\mathrm{i} = \mathrm{T_c}$ must be fulfilled for the condition of coordination of both SPs.

Thus, the translation coordinates can be mapped as follows. The tool-SP moving platform translation follows the same derivation as in the single configuration (Equation 6.9). However, the tool contact point ${}^{\mathrm{F}}\mathrm{T_c}$ (x',y',z') must be calculated from the input coordinates ${}^{\{P2\}}\boldsymbol{X}_{\mathrm{i}}$. In order to determine ${}^{\mathrm{F}}\mathrm{T_c}$ (x',y',z'), the translation input coordinates x,y,z can be transformed with respect to frame $\{F\}$. According to Figure 6.5, the tool contact point with respect the table-SP base frame $\{F2\}$ is the addition of two position vectors in Equation 6.13, where ${}^{F2}\boldsymbol{R}_{P2}$ is the rotation matrix of frame $\{P2\}$ relative to frame $\{F2\}$.

$$ {}^{\{F2\}}\boldsymbol{X}_{\mathrm{i}} = \begin{bmatrix} q_{x2} \\ q_{y2} \\ q_{z2} \end{bmatrix} + {}^{F2}\boldsymbol{R}_{P2}\,{}^{\{P2\}}\boldsymbol{X}_{\mathrm{i}} = \begin{bmatrix} q_{x2} \\ q_{y2} \\ q_{z2} \end{bmatrix} + {}^{F2}\boldsymbol{R}_{P2}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.13} $$

This rotation matrix can be derived in the following manner. Since the relationship of the base frame $\{F\}$ and $\{F2\}$ is known in Equation 6.11,

$$ {}^{\mathrm{F}}\mathrm{T_c} \equiv \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = {}^{\{F2\}}\boldsymbol{X}_{\mathrm{i}} + \begin{bmatrix} 0 \\ 0 \\ -h2 \end{bmatrix} = \begin{bmatrix} q_{x2} \\ q_{y2} \\ q_{z2} \end{bmatrix} + {}^{F2}\boldsymbol{R}_{P2}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -h2 \end{bmatrix} \tag{6.14} $$

The table-SP moving platform has three extra DOF. As can be inferred from Equation 6.14, although the input coordinates are defined relative to

frame {*P2*}, the translation of the moving platform is not restricted as long as the tool-SP moves the tool such that its tip touches the tool contact point ${}^{F}T_{c}$.

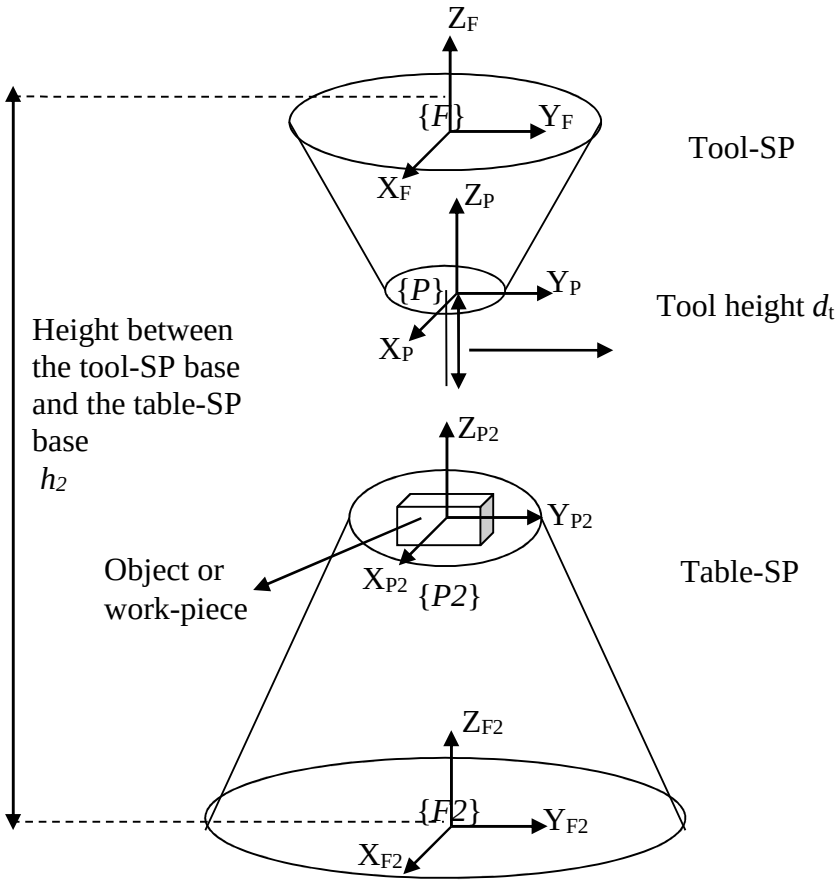


Figure 6.4 Coordinate system in extended configuration

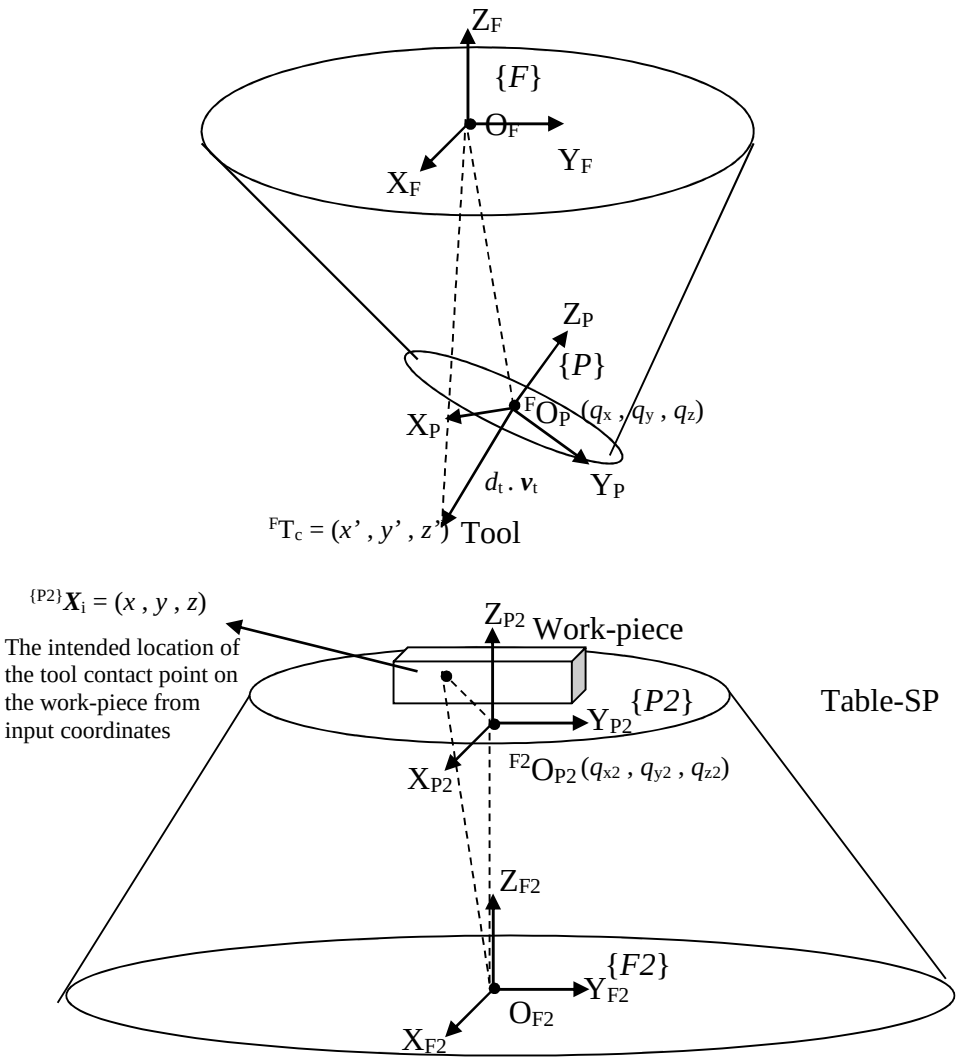


Figure 6.5 Coordination of the tool-SP and the table-SP to reach the input coordinate (the tool contact point on the work-piece)

Rotation coordinates mapping is more complicated if the sequence of the rotation is arbitrary. However, the tool-SP has two rotation DOFs that are used for machining and the third DOF is not used because it rotates about the tool axis. This is also true for the table-SP. To simplify rotation coordinates mapping, the rotation axes of the table-SP are made the same with the

rotation-axes of the tool-SP, so that the angles can be added arithmetically. The first two rotation coordinates of the tool-SP are used for machining, as the last one is a free DOF that can be set arbitrarily. It can also be set for the table-SP that the first two rotations are the DOFs that are used in machining. The two rotation axes (Φ, Γ) are mapped to the four rotation DOFs of both tool-SP and table-SP (θ_1,θ_2,θ_{12},θ_{22}), one in each couple is being redundant to one another (i.e. θ_1 is redundant to θ_{12} and θ_2 is redundant to θ_{22}).

Figure 6.6 describes the rotation axes used in the tool-SP in the single configuration. The table-SP Euler angles must have the same convention as the Euler angles of the tool-SP so that the angle can be combined. The table-SP first rotation is along the Z_{P2} axis, which is parallel to the Z_P axis of the tool-SP. So, the first input rotation coordinate can be mapped using Equation 6.15..

$$\Phi = \theta_1 - \theta_{12} \tag{6.15}$$

The minus sign in Equation 6.15 comes from the fact that the table-SP locates the work-piece relative to the tool, so it becomes the opposite direction. Moreover, the rotation matrix for the first rotation on the table-SP can be calculated using Equation 6.16.

$$\boldsymbol{R}_{\theta 12} = \begin{bmatrix} C_{\theta 1} & -S_{\theta 1} & 0 \\ S_{\theta 1} & C_{\theta 1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.16}$$

The second rotation axis of the table-SP is defined as parallel to the second rotation axis of the tool-SP, so that the angle can also be summed using Equation 6.17.

$$\Gamma = \theta_2 - \theta_{22} \tag{6.17}$$

The second rotation axis is along the new axis X_P after the first rotation on the tool-SP. The rotation sequence on the table-SP can be summarized in Figure 6.7. The rotation matrix for the second rotation on the table-SP can be constructed by using axis-angle representation of the rotation about the vector X_P'. The vector X_P' is the first column of matrix $\boldsymbol{R}_{\theta 12}$.

$$\boldsymbol{R}_{\theta 22} = \begin{bmatrix} S^2_{\theta 1}(1-C_{\theta 22})+C_{\theta 22} & -S_{\theta 1}C_{\theta 1}(1-C_{\theta 22}) & C_{\theta 1}S_{\theta 22} \\ -S_{\theta 1}C_{\theta 1}(1-C_{\theta 22}) & C^2_{\theta 1}(1-C_{\theta 22})+C_{\theta 22} & S_{\theta 1}S_{\theta 22} \\ -C_{\theta 1}S_{\theta 22} & -S_{\theta 1}S_{\theta 22} & C_{\theta 22} \end{bmatrix} \tag{6.18}$$

Although it is possible to define the third rotation, the use of the DOF may not be necessary for the current machining application. Therefore, the third rotation is fixed at 0, or it can be defined that the rotation matrix of the third rotation on the table-SP is an identity matrix.

$$\boldsymbol{R}_{\theta 32} = \boldsymbol{I} \tag{6.19}$$

The rotation matrix of frame {*P2*} relative to frame {*F2*} is the multiplication of the above rotation matrices according to the rotation sequence.

$${}^{F2}\boldsymbol{R}_{P2} = \boldsymbol{R}_{\theta 12}\,\boldsymbol{R}_{\theta 22}\,\boldsymbol{R}_{\theta 32} \tag{6.20}$$

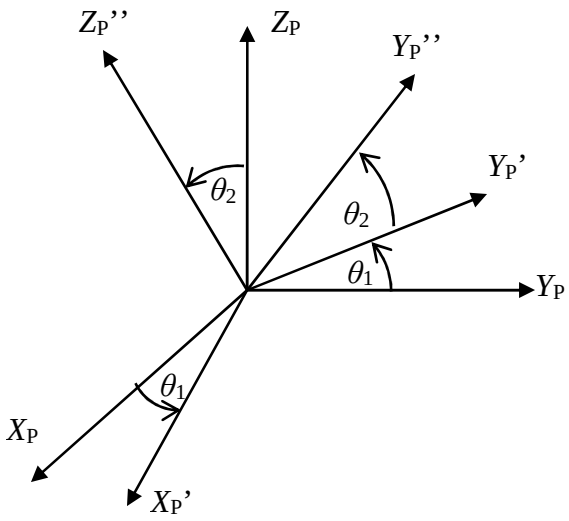


Figure 6.6 Rotation sequence in the tool-SP

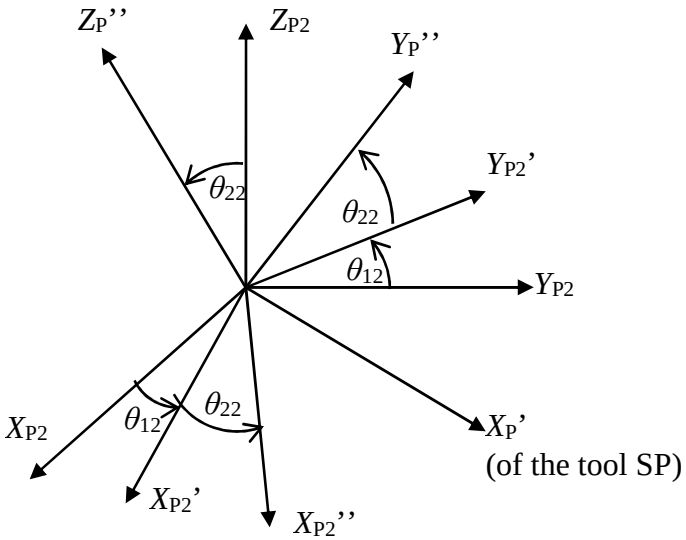


Figure 6.7 Rotation sequence in the table-SP

Coordinate mapping can be summarized in Figure 6.8.

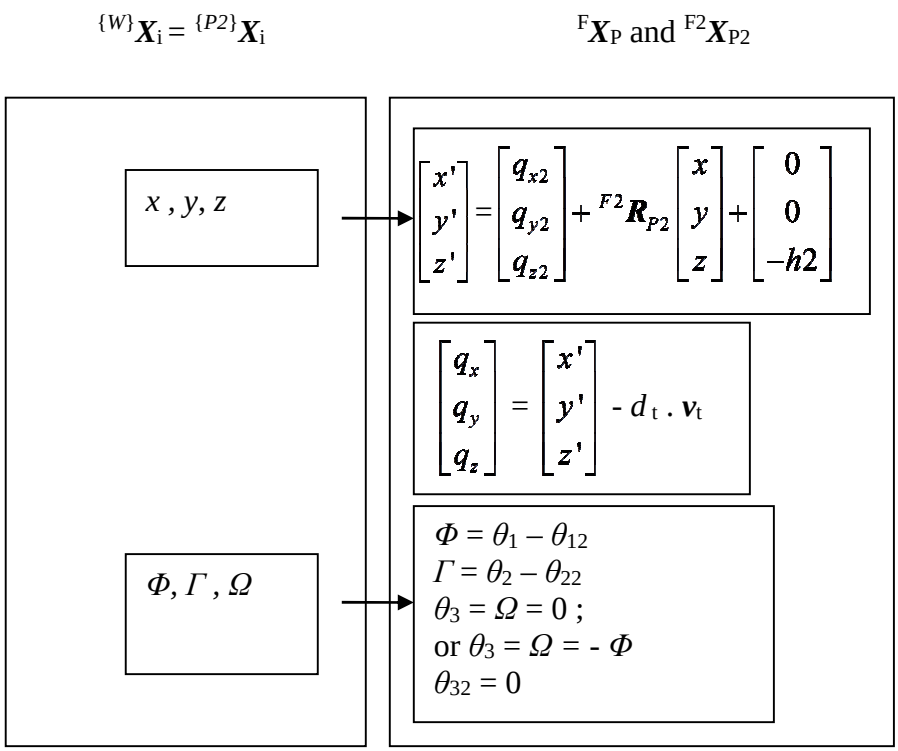


Figure 6.8 Coordinate mapping in extended configuration

6.2 Extended Configuration Motion Planning

This section addresses the problem of optimal trajectory planning for the cooperative manipulators. The objectives are to maximize the stiffness of the structure and minimize the force requirement of the actuators while satisfying the constraints of workspace and singularity. A computational algorithm for searching the optimum location of machining trajectory is developed. The performance index is based on the condition number of the Jacobian matrix of the Stewart platforms.

The condition numbers are accepted as a measure of the manipulator performance as the Jacobian matrix represents the relationships between the velocities and the force acting on the actuators, and the velocities and the twist

of the end-effector (Stoughton and Arai 1993). If the Jacobian matrix becomes singular (i.e., when the condition number reaches infinity), the manipulator will gain additional freedom and lose its rigidity and ability to move and apply forces in arbitrary directions. The condition number expresses the amplification of the relative error in the joint coordinates to the error of the moving platform pose. Thus, the condition number has to be as small as possible. The following are the derivation of the Jacobian matrix of the Stewart platform and the corresponding condition numbers.

6.2.1 Jacobian Matrix and Condition Number

For a PKM, the Jacobian matrix is defined as

$$\dot{\boldsymbol{l}} = \boldsymbol{J(X)}\ \dot{\boldsymbol{X}} \tag{6.21}$$

where $\dot{\boldsymbol{l}}$ is the rate of change of the length of the legs and $\dot{\boldsymbol{X}} = [\dot{q}^T\ \dot{\theta}^T]^T = \left[\dot{q}^T\ \boldsymbol{\omega}^T\right]^T$ is the Cartesian velocity of the end effector. Using inverse rate kinematics (Harib and Srinivasan 2003), both Jacobian matrices can be obtained. The location of the ball-socket joints are differentiated with respect to time, and the projections of these velocity vectors on the axis of the legs yield the extension rates of legs of the tool-SP and the table-SP as in the following.

$$\dot{l}_i = {}^F\dot{a}_i \cdot n_i = {}^F\dot{q}_P \cdot n_i + \boldsymbol{\omega} \cdot \left({}^F\mathbf{R}_P \cdot {}^P a_i\right) \times n_i\ ,\ \boldsymbol{\omega} = \left[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3\right]^T ; i = 1,2,..6 \tag{6.22}$$

$$\dot{l}_{i2} = {}^{F2}\dot{a}_{i2} \cdot n_{i2} = {}^{F2}\dot{q}_{P2} \cdot n_{2i} + \boldsymbol{\omega}_2 \cdot \left({}^{F2}\mathbf{R}_{P2} \cdot {}^{P2}a_{i2}\right) \times n_{i2}\ ;\ \boldsymbol{\omega}_2 = \left[\dot{\theta}_{12}, \dot{\theta}_{22}, \dot{\theta}_{32}\right]^T ;$$

i = 1,2,..6 (6.23)

The Jacobian matrices can be obtained through comparing the following equations derived from Equation (6.21).

$$\dot{\boldsymbol{l}} = \mathbf{J}\ \dot{\boldsymbol{X}}\ = \mathbf{J}\begin{pmatrix} {}^{F}\dot{q}_{P} \\ \boldsymbol{\omega} \end{pmatrix} \tag{6.24}$$

$$\dot{\boldsymbol{l}}_{2} = \mathbf{J}_{2}\dot{\boldsymbol{X}}_{2} = \mathbf{J}_{2}\begin{pmatrix} {}^{F2}\dot{q}_{P2} \\ \boldsymbol{\omega}_{2} \end{pmatrix} \tag{6.25}$$

Thus, the Jacobian matrix for the tool-SP and the table-SP are

$$\mathbf{J} = \begin{bmatrix} n_{1}^{T} & \left({}^{F}\mathbf{R}_{P} \cdot {}^{P}a_{1} \times n_{1}\right)^{T} \\ n_{2}^{T} & \left({}^{F}\mathbf{R}_{P} \cdot {}^{P}a_{2} \times n_{2}\right)^{T} \\ \vdots & \vdots \\ n_{6}^{T} & \left({}^{F}\mathbf{R}_{P} \cdot {}^{P}a_{6} \times n_{6}\right)^{T} \end{bmatrix} \tag{6.26}$$

$$\mathbf{J}_{2} = \begin{bmatrix} n_{12}^{T} & \left({}^{F2}\mathbf{R}_{P2} \cdot {}^{P}a_{12} \times n_{12}\right)^{T} \\ n_{22}^{T} & \left({}^{F2}\mathbf{R}_{P2} \cdot {}^{P}a_{22} \times n_{22}\right)^{T} \\ \vdots & \vdots \\ n_{62}^{T} & \left({}^{F2}\mathbf{R}_{P2} \cdot {}^{P2}a_{62} \times n_{62}\right)^{T} \end{bmatrix} \tag{6.27}$$

The condition number, κ, of the Jacobian matrix **J** is defined as (Golub and Van Loan 1983),

$$\kappa = \left\|\boldsymbol{J}\right\|\left\|\boldsymbol{J}^{-}\right\| \tag{6.28}$$

$$\kappa_{2}\left(\boldsymbol{X}_{2}\right) = \left\|\mathbf{J}_{2}\right\|\left\|\mathbf{J}_{2}^{-}\right\| \tag{6.29}$$

where $\|\boldsymbol{J}\|$ denotes the 2-norm of the Jacobian matrix **J**. Since the Jacobian matrix is a function of the moving platform pose, the condition number is also a function of the pose.

The condition number will be inconsistent if it is derived directly from the non-homogeneous Jacobian matrix **J**. To avoid this problem, the rotational elements, i.e., the last three columns of the Jacobian matrices are divided by the radius of the mobile platform of each manipulator (Ma and Angeles 1991b) before the computation of the condition number.

In the next section, the condition number is used in the objective function in the optimization procedure to solve the redundant parameters. However, as there are two Stewart platforms in the cooperative manipulators, an objective function that multiplies the condition numbers of both platforms is defined.

$$f\left(\boldsymbol{X}_1, \boldsymbol{X}_2\right) = \kappa_1\left(\boldsymbol{X}_1\right) \cdot \kappa_2\left(\boldsymbol{X}_2\right) \tag{6.30}$$

The condition number ranges from 1.0 to infinity, i.e., $1 < \kappa < \infty$. Thus, the value of the scalar multi-variable function *f* has the same range and it reflects the performance of both Stewart platforms with respect to their particular pose. Through minimizing this function, the optimal pose configuration can be obtained for a given input.

For stiffness analysis, the stiffness measure is obtained by computing the stiffness matrix which is given by Equation 6.31.

$$\vartheta = \boldsymbol{J}^T \chi \boldsymbol{J} \tag{6.31}$$

where χ = diag [k_1,k_2,…,k_6] and k_1,k_2,…,k_6 are the axial stiffness of the legs (El-Khasawnath and Ferreira 1999). The stiffness of the SP is taken as the minimum eigenvalue of matrix ϑ. The optimization will yield trajectory which direction is much affected by the maximum mean of the stiffness value.

In addition, in the optimization singularity must be avoided. It can be formulated for each SP in the following. If the actuated leg lengths are denoted by $\boldsymbol{l}$ and the location of the end-effector is denoted by $\boldsymbol{X}$, then the singularity constraint imposed on the SP is

$$\boldsymbol{J}\,\dot{\boldsymbol{X}} = \dot{\boldsymbol{l}} \tag{6.32}$$

Conceptually, if the $\boldsymbol{J}$'s determinant is equal to zero, the corresponding condition number will reach an infinite value and thus, infinitesimal motion of the end-effector can exist although all leg lengths are fixed.

6.2.2 Optimization Procedure

Optimization (minimization) of the condition number is carried out to determine the redundant parameters so that the resulting pose will give the best dexterity for a given task. The reachable workspaces of both platforms need to be validated prior to the minimization to ensure that the solution is attainable in practice. The kinematic constraints that limit the Stewart platform workspaces have been described in Section 5.1. In addition, the optimization to determine the pose of both moving platforms is according to the coordinate mapping in the extended configuration in Section 6.1.2. The

optimization will solve the coordinates of the moving platforms, ${}^{F}\boldsymbol{X}_{P} = [q_{x}, q_{y}, q_{z}, \theta_{1}, \theta_{2}, \theta_{3}]^{T}$ for the tool-SP and ${}^{F2}\boldsymbol{X}_{P2} = [q_{x2}, q_{y2}, q_{z2}, \theta_{12}\ \theta_{22}, \theta_{32}]^{T}$ for the table-SP, given an input coordinate ${}^{\{P2\}}\boldsymbol{X}_{i} = [x, y, z, \Phi, \Gamma, \Omega]^{T}$. In the coordinate mapping, the value of θ_{32} is equal to zero, and the value of θ_{3} can be chosen to be zero or - Φ. Therefore, these two coordinates are fixed and do not change during the optimization.

Simulation is carried out in MatLab using the *fmincon* function in the MatLab Optimization Toolbox. The *fmincon* function finds a constrained minimum of a scalar function of several variables with an initial guess and any number of inequalities or equations. This is generally known as constrained nonlinear optimization. This function is used to solve the coordinates of the moving platforms with the minimum objective value as defined in Equation 6.30. An additional constraint is added to the algorithm so that the resulting trajectory is smooth. The algorithm will search for the optimum pose with the lowest condition number, but within the adjacent range from the previous pose. It is done by adding the following constraints.

$$\begin{aligned} \left({}^{F}\boldsymbol{X}_{P}{}^{i-1} - \Delta\boldsymbol{X}\right) \ &\leq\ {}^{F}\boldsymbol{X}_{P}{}^{i} \ \leq\ \left({}^{F}\boldsymbol{X}_{P}{}^{i-1} + \Delta\boldsymbol{X}\right) \\ \left({}^{F2}\boldsymbol{X}_{P2}{}^{i-1} - \Delta\boldsymbol{X}\right) \ &\leq\ {}^{F2}\boldsymbol{X}_{P2}{}^{i} \ \leq\ \left({}^{F2}\boldsymbol{X}_{P2}{}^{i-1} + \Delta\boldsymbol{X}\right) \end{aligned} \tag{6.33}$$

where i denotes the step number.

By limiting the range of the next pose, the trajectory will not change drastically from one point to another point. The *fmincon* function only estimates the local minimum within a certain range. If the first guess for all the trajectory points in the optimization is set as a constant, there is a chance that the solution will not converge to the target range specified. Therefore, the

previous pose is used as the first guess for the next computation. The computation will begin from a constant guess only for the first point of the trajectory. Figure 6.9 summarizes the algorithm.

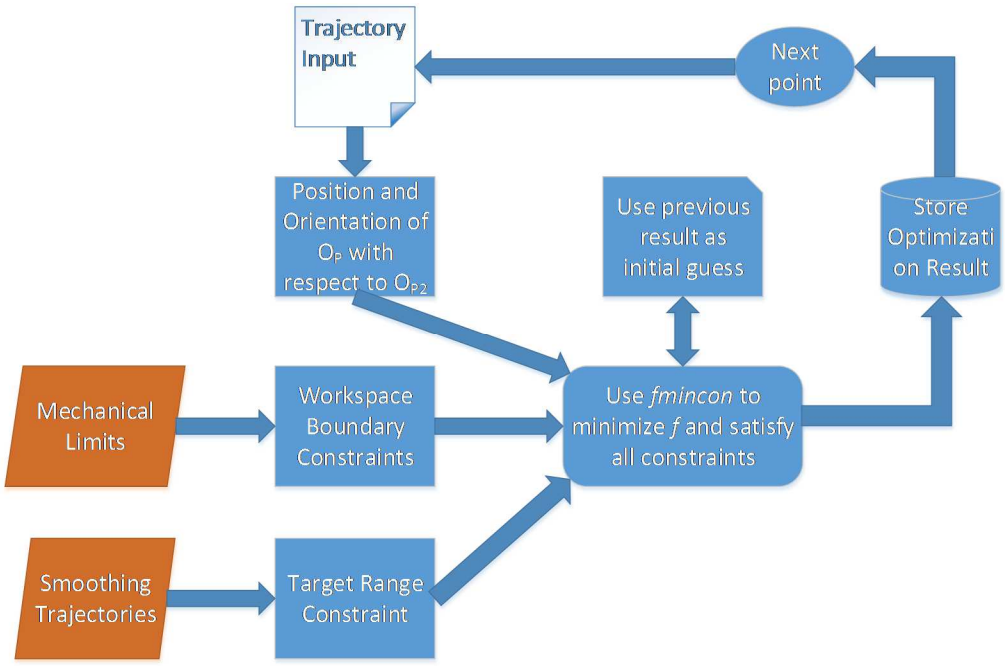


Figure 6.9 Algorithm for motion planning in extended configuration

6.2.3 Straight-line Milling

The proposed algorithm is tested with a sequence of ${}^{\{P2\}}\boldsymbol{X}_{\mathrm{i}}$ corresponding to an arbitrary linear input trajectory. The values used in this simulation are shown in Table 6.1. In addition, since the *fmincon* function is based on a numerical approach, a first guess is required. In this simulation, the first guess used was $q_z = q_{z2} = Z_0/2$, and the other parameters are zero ($q_x = q_{x2} = x_0 = 0$; $q_y = q_{y2} = y_0 = 0$).

The first simulation is done without the smoothing constraint. The result is shown in Figure 6.10 for q_z. Although the proposed algorithm had

successfully solved the trajectory, the result is not satisfactory. From the figure, it can be seen that there are discontinuities in these solutions. The results of other coordinates also show similar characteristics. With the additional smoothing constraints, the optimization with the same input was repeated. The results are shown in Figure 6.11. No discontinuity was observed, such that the actual motion based on this trajectory will be smooth. Hence, with the added constraints, this trajectory planning becomes practical and further filtering can be employed when a better trajectory is required. In addition, a comparison of the trajectory with and without smoothing constraints is shown in Table 6.2.

Table 6.1 Values used in the example

First guess value (q_z, q_{Z2}, θ_1, θ_2, θ_{12}, θ_{22})		250 mm , 250 mm , 0°, 0°, 0°, 0°
Smoothing constraint		
	$\Delta \boldsymbol{X}$	(1 mm, 1mm, 1mm, 0.25° , 0.25° , 0.25°)
Input ${}^{\{F2\}}\boldsymbol{X}_i$ (X, Y, Z, Φ, Γ, Ω)		
	Start pose	10 mm , 10 mm , 1 mm, 5°, 5°, 5°
	End pose	10 mm , 10 mm , 50 mm, 5°, 5°, 5°
Z_0		500 mm
x_0 , y_0		0 mm , 0 mm

Table 6.2 Trajectory Planning Result Summary

	Without Smoothing Constraints	With Smoothing Constraints
Δq_x		
Mean (mm)	1.456	0.213
Standard Deviation	0.73	0.61
Mean of first derivative (mm/s)	0.567	0.342
Standard Deviation of first derivative	0.63	0.43

Δq_y		
Mean (mm)	2.023	0.145
Standard Deviation	0.58	0.45
Mean of first derivative (mm/s)	0.675	0.312
Standard Deviation of first derivative	1.13	0.87
Δq_z		
Mean (mm)	0.678	0.087
Standard Deviation	1.3	0.72
Mean of first derivative (mm/s)	2.134	1.536
Standard Deviation of first derivative	0.345	0.287

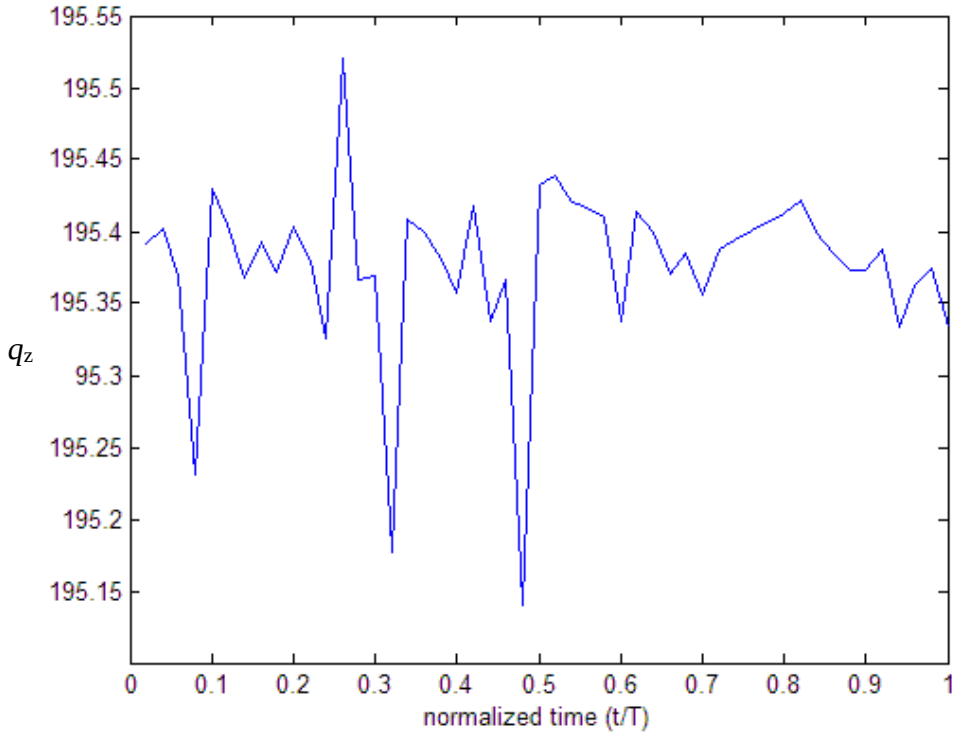
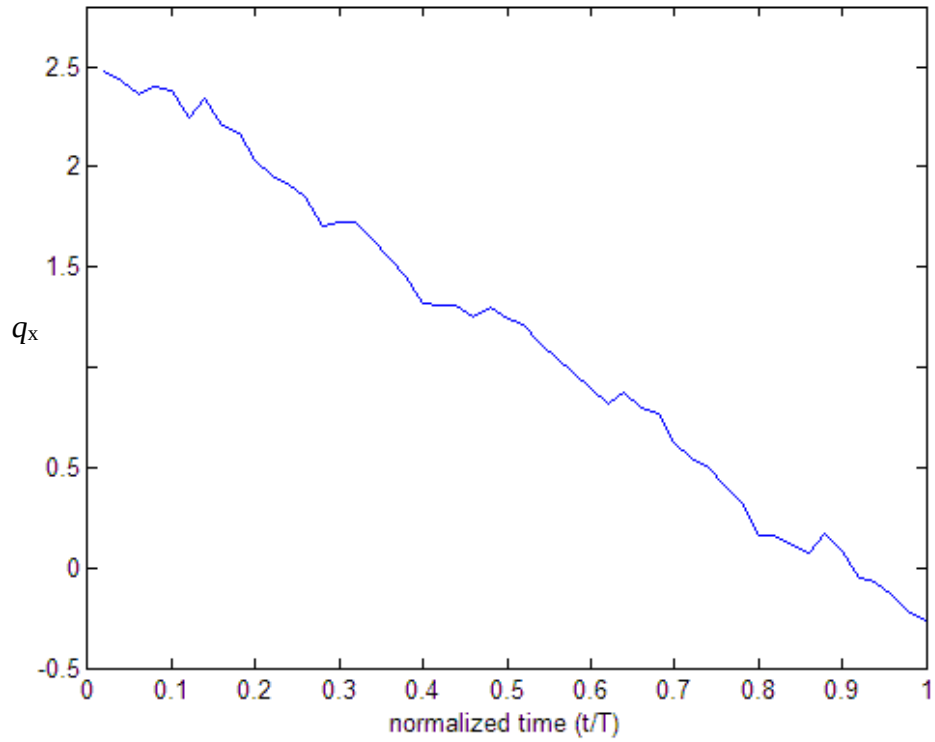
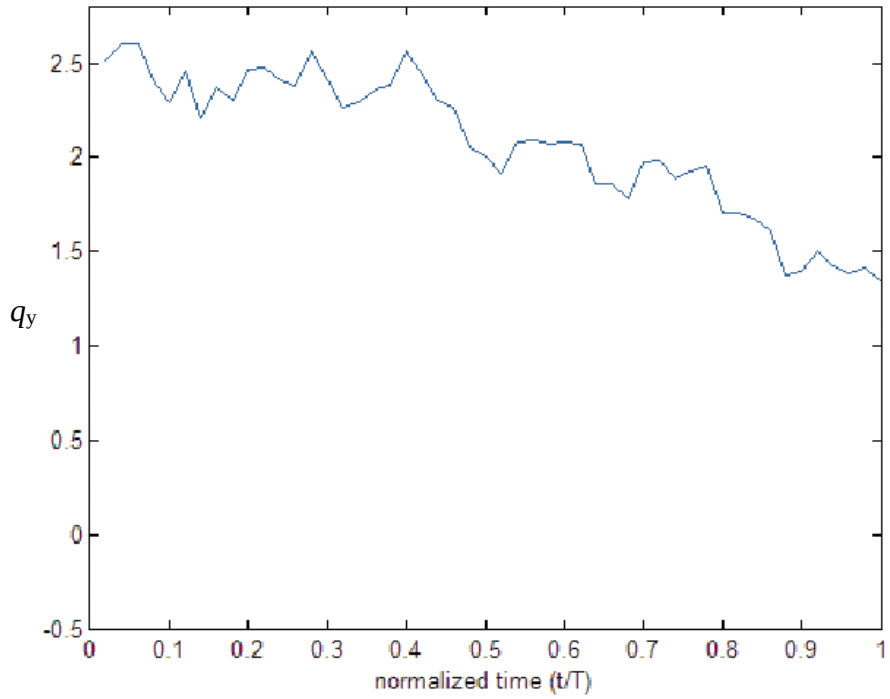


Figure 6.10 Resulting trajectory plan q_z from the optimization procedure

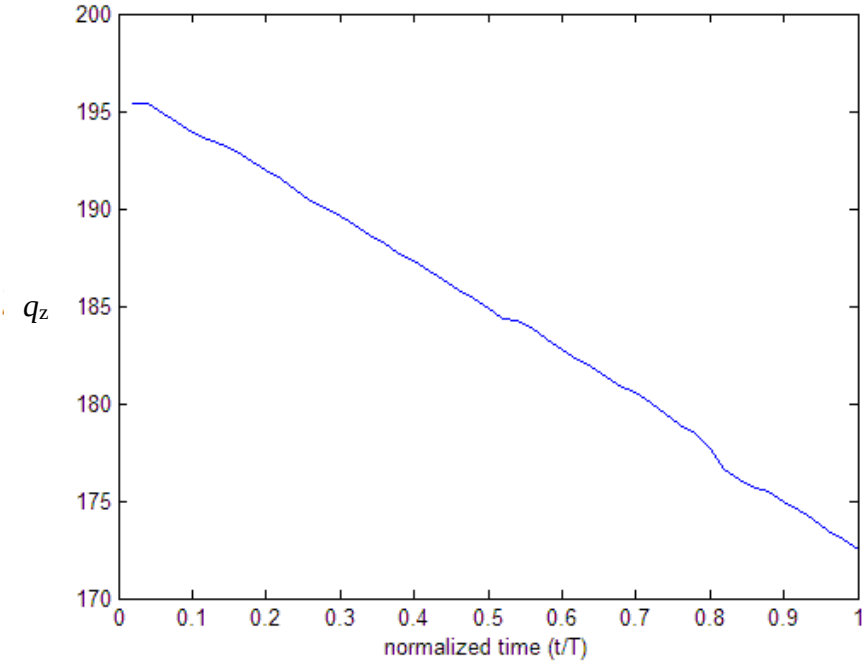
(t = step number, T = total number of steps = 50)


2.5
2
1.5
q_x
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
normalized time (t/T)

(a)


2.5
2
1.5
q_y
1
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
normalized time (t/T)

(b)



(c)

Figure 6.11 A smooth trajectory from the algorithm with extra constraint; showing: (a) q_x, (b) q_y, (c) q_z (in mm, t = step number, T = total number of steps = 50)

6.3 Stewart Platform Machining Framework with CAD/CAM Software

Figure 6.12 shows a flowchart of how a machining job can be planned and conducted for the cooperative manipulators. The arrangement of the cooperative manipulators has been presented. The tool-SP and the table-SP are anchored in a common frame. This is an arrangement so that both platforms can work cooperatively to enlarge effective workspace. However, each Stewart platform can be utilized independently or cooperatively. There are three options that can be done with the cooperative manipulators:

1. Only the tool-SP moves with the cutting tool, the table-SP is stationary. This is the same as the single configuration.
2. Only the table-SP moves with the work-piece, the tool-SP is stationary. The table-SP must perform the motion in reverse to the tool motion to achieve the desired result.
3. Both Stewart platforms move cooperatively. The motion of both platforms must be coordinated. They can move one at a time or both at the same time. This is the same as the extended configuration.

In this research, the second option is not used. This is due to the design of the table-SP that has a larger workspace but lower accuracy. To maintain high accuracy and continuance in the machining operation, both Stewart platforms move at different timings. This is due to the observation that the results are not satisfactory because the error is too high when both SP moves at the same time during cutting. Tool-SP is used to perform the material removal process, while the table-SP is to move the work-piece while not cutting.

A fixture was specially designed to hold a work-piece of a maximum 90 mm by 90 mm by 70 mm on top the tool Stewart Platform or the machining table (in the single configuration). The fixture consists of an L-shape locator, which provides the reference point, and two clamps which provide the clamping force. Clamps are fixed to the platform through two screws. A slot is designed in each clamp, hence the position of the screw is adjustable and the fixture will be able to accommodate various work-piece sizes within the limit. The fixture can be seen in Figure 6.13.

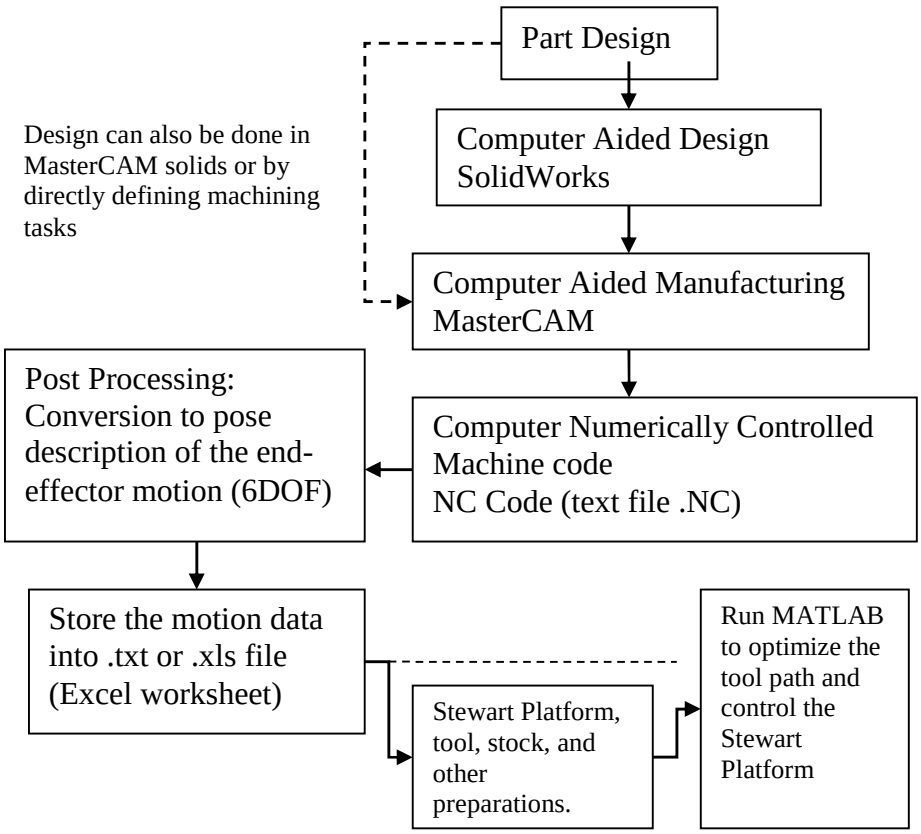


Figure 6.12 Steps to machining with the cooperative manipulators

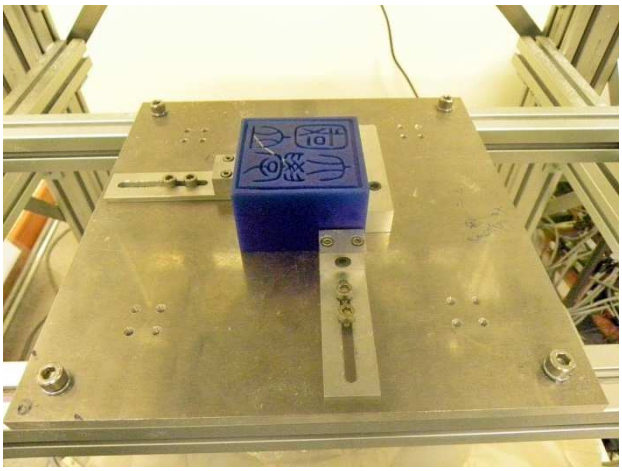


Figure 6.13 Fixture for holding work-piece on top of the tool-SP

A part can be designed using a CAD software. In this research, a CAD software SolidWorks is used to design and model the mechanical objects. In addition, a CAM software, MasterCAM is used to generate the necessary machining tool path (NC code) to remove the material from a stock, so that it becomes shaped as designed in CAD. There are two ways of creating a design and generating the corresponding machining NC codes as shown in Figure 6.14.

a) Part geometry is defined in CAD (SolidWorks) and then it is saved as Parasolid or MasterCAM Solid Feature Binary (.SFB) format to be imported to CAM (MasterCAM). The tool path will be generated in MasterCAM.
b) Part geometry is defined as solids in CAM (MasterCAM solids). The tool path will be generated in MasterCAM.

The first approach is easier when the user is more familiar with SolidWorks or other CAD software that can export Parasolid format. Because an intermediate data file is used, some design features may not be fully compatible or recognized by MasterCAM. The second approach is recommended if the user is familiar with the modelling of the part directly in CAM. This approach will not have any incompatibility issues as all the processes are done in a single software.

In the next few sections, the first approach will be used, but with more advanced integration between SolidWorks and MasterCAM for user- friendly and easier procedure. This method uses an extension of MasterCAM and SolidWorks, so that the part design and tool path generation can be done in

the SolidWorks user interface. This method can only process up to 3-axis machining jobs. For 5-axis machining, the second approach must be used. In addition, Figure 6.15 shows machining parameters that can be set for a machining job in a CAM software.

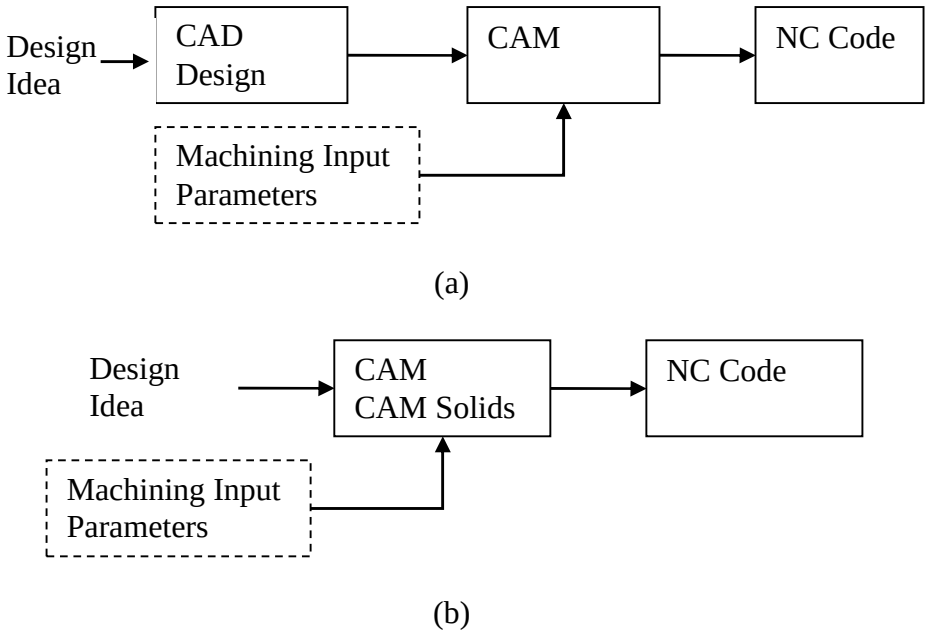


Figure 6.14 Information flow of part design and NC code generation

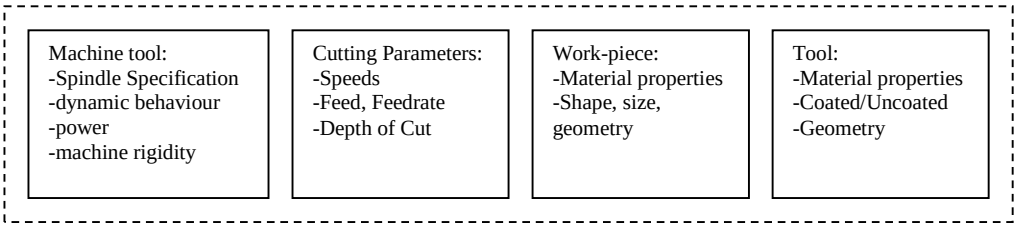


Figure 6.15 Machining Input Parameters

6.3.1 Tool Path Post-processing

In order to generate the tool path from the NC codes generated from MasterCAM, it is necessary to go through post-processing, which acts as a

translator that reads the tool path instructions issued from MasterCAM and writes an appropriate trajectory for the Stewart platform. The post-processor is written in MatLab.

The MatLab program does not need commentaries or additional information which is usually included in the NC file. In order to convert the NC file to a compatible format, the first few lines up which contain additional information till the line before ‘G21’ should be deleted as shown in Figure 6.16 in blue colour. Besides that, the block numbers should be removed by choosing the menu ‘NC Functions’, and select ‘Remove Block Numbers’ similarly shown in Figure 6.17.

```
:0001 ( PROGRAM – TEST13 )
( DATE – 13-03-12 TIME – 17:45 )
G21
G0 G17 G40 G80 G94 G98
G0 G28 G91 Z0.
( 3. FLAT END MILL HSS TOOL -1 DIA. OFF. – 1. )
( WCS: TOP )
T1 M6
G0 G54 G90 X18.506 Y25.002 C89.851 s2546 M3
…
```

Figure 6.16 Additional information in a G-Code file that cannot be processed by the MatLab post processor (blue codes)

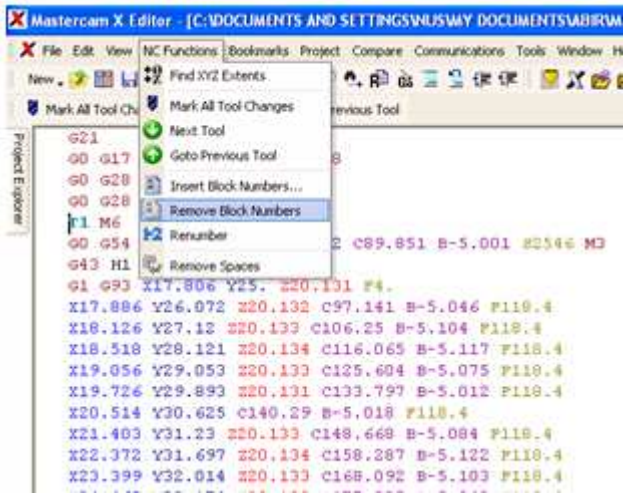


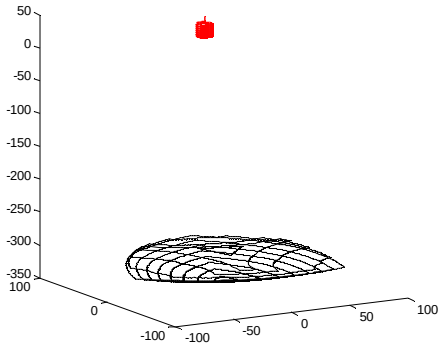
Figure 6.17 Block Numbers removal in the NC code

The MatLab post-processor converts the NC codes to the input coordinates for the cooperative manipulators according to the coordinate mapping defined in Section 6.1. There are two steps in conducting the MatLab post-processor. First, the program will scan the NC file and extract the G-codes and motion types.

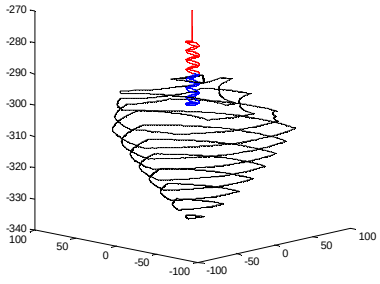
Secondly, the cutter locations that will be input to the MatLab program for moving the Stewart platform must be able to be interpreted by the Stewart platform. This process includes extraction and interpolation of Cutter Location (CL) Data. This process continues from the previous step, where the via points are inserted in between two subsequent CLs with a predetermined interpolation step.

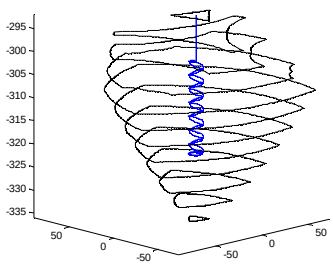
Furthermore, before conducting any machining, a test on the tool path must be performed to check whether all the points are reachable to ensure proper operation. The workspace simulation can be superimposed here to result in a plot such as shown in Figure 6.18. Accessible trajectory CL points

are plotted continuously in blue colour. If some parts of the simulated trajectory are plotted in red colour, then these parts are not accessible by the Stewart platform. In the case that the whole trajectory has filled the entire Stewart platform workspace, the machine origin must be fine-tuned to machine this trajectory in such a way that the trajectory starting point is near the boundary, so that the trajectory can be covered by the workspace as shown in Figure 6.18c.



(a)





(b) (c)

Figure 6.18 Testing feasibility of the resulting trajectory, (a) Inaccessible trajectory (b) Partially accessible trajectory (c) Accessible trajectory

6.3.2 Determining Machine Origin

The machine origin is defined as the position (0,0,0) of the tool when all the Stewart platform joint coordinates are zeroed (home position). This is used as the reference for determining the start point of the trajectory. The start point may not be at the machine origin, but its location is determined with respect to the machine origin. The procedure to determine the machine origin is done manually. In the single configuration, the tool-SP is moved to its home position. Next, the distance along the base Z_F axis between the tool tip and the machining table (Z_distance) is measured (Figure 6.19). The machine origin is then simply given by the following equations.

X_Machine_origin = X_Current_Cutter_Location;

Y_Machine_origin = Y_Current_Cutter_Location;

Z_Machine_origin = Z_Current_Cutter_Location – Z_distance; (6.34)

In the extended configuration, similar procedures can be used. The tool-SP and the table-SP are configured in their home position before the procedure is conducted. The distance between the tool- and the table-SP moving platform is measured.

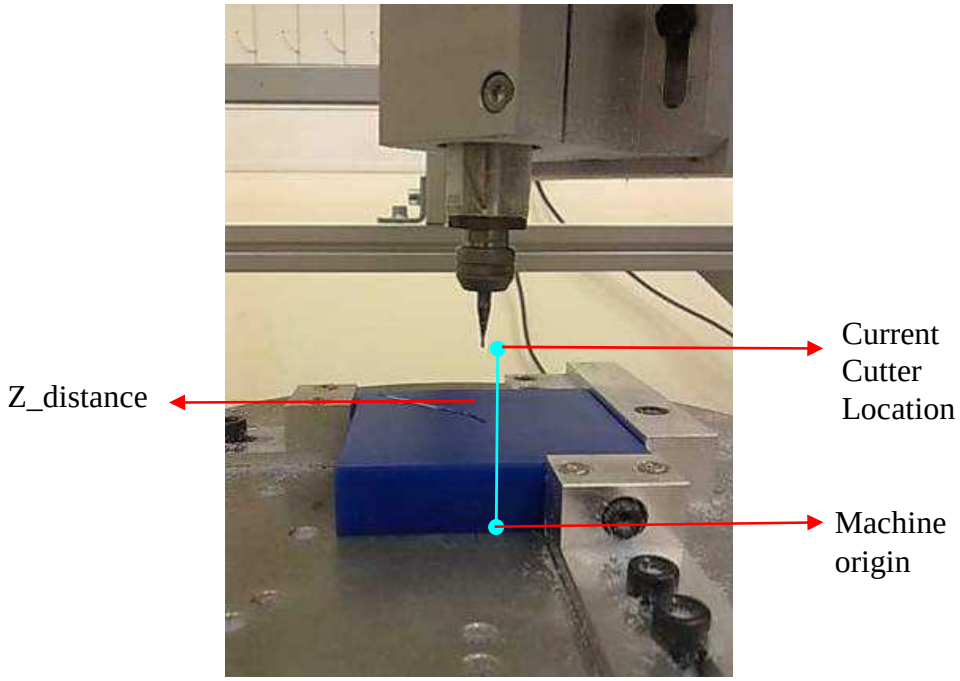


Figure 6.19 Determining the machine origin

6.4 Machining Case Studies

The parts to be machines are designed in SolidWorks which is a Parasolid-based solid modeller, and utilizes a parametric feature-based approach to create models and assemblies. Parameters refer to constraints which values determine the shape or geometry of the model or assembly. Parameters can be either numeric parameters, such as line lengths or circle diameters, or geometric parameters, such as tangent, parallel, concentric, horizontal or vertical, etc.

6.4.1 Machining an 'NUS' Pocket

After the stock has been defined, a sketch is created on the top surface (Figure 6.20). The Extruded Cut feature is used to generate the pocket. The tool path is generated using the MasterCAM Feature Based tool path function

that will recognize any feature after the stock dimension has been defined. The cutter diameter is also selectable in the interface. Each tool path group can be exported to a separate NC file if needed, especially when machining curve or complex surfaces with a large number of data points, because one whole chunk of the tool paths will involve huge amount of memory for processing in MatLab and may halt the Stewart platform in the middle of an operation. The resulting trajectory and the actual machining are shown in Figure 6.21.

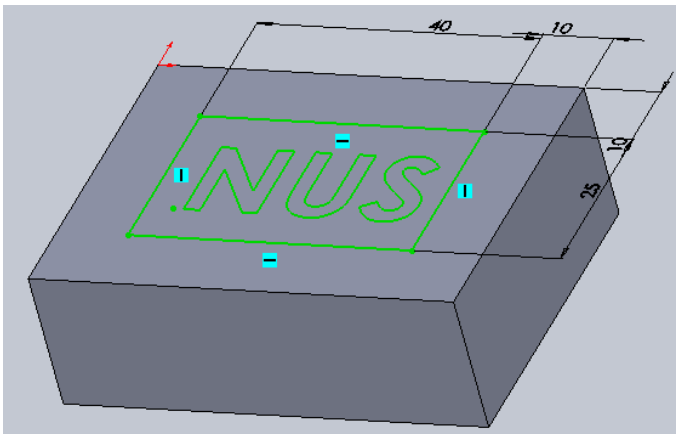
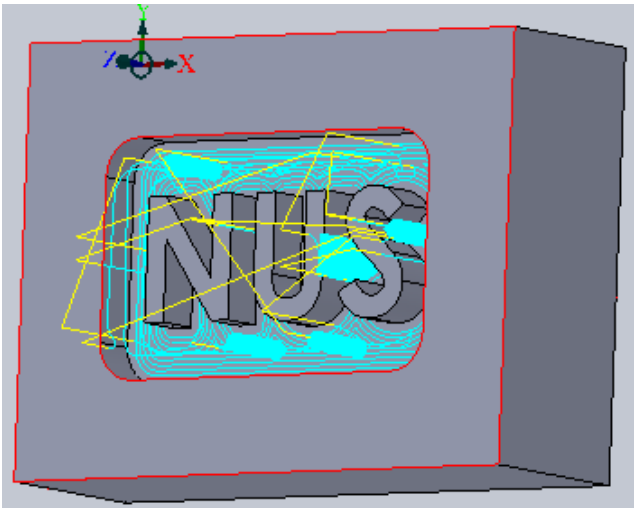
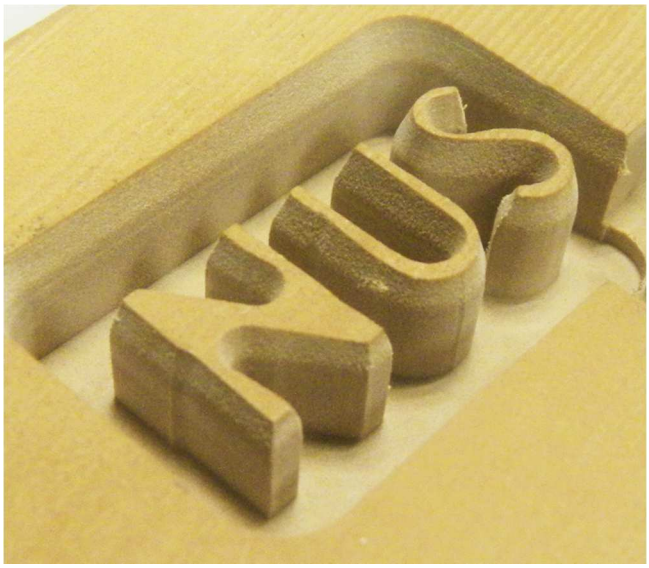


Figure 6.20 Sketch for the ‘NUS’ pocket on top of the stock model



(a)



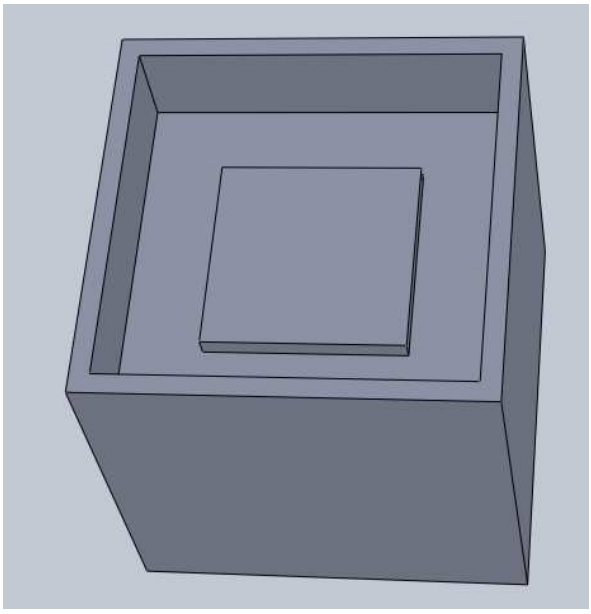
(b)

Figure 6.21 (a) The tool-path generated (b) Actual machining

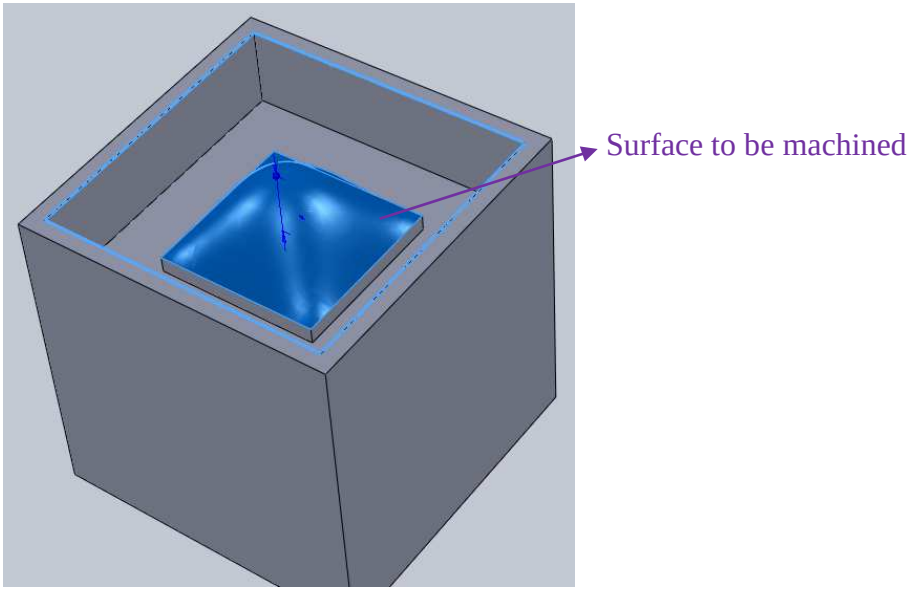
6.4.2 Machining a Dome

Using SolidWorks cut extrude followed by boss extrude will form a base as shown in Figure 6.22a. A dome-shaped island is added on top of the rectangular base. The machining is done in two stages: roughing and finishing. Accordingly, the MasterCAM generates one tool path group for each tool path

group. The roughing creates a pocket with a rectangular island. The finishing produces a tool path trajectory with high density of coordinates to machine the dome shape (Figure 6.23). The finishing tool path is generated based on the dome surface, which is selectable in the interface (Figure 6.22b). In addition, the tool path must be contained in the pocket to make sure no tool path is generated outside the pocket. The tool containment is defined as a closed curve in Figure 6.24.

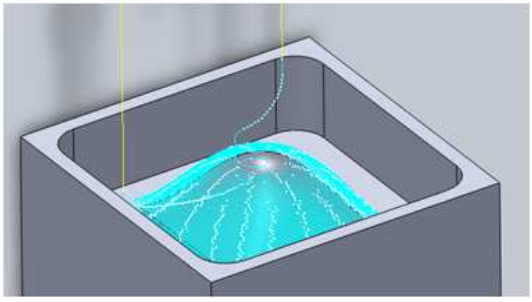


(a) Pocket with an island

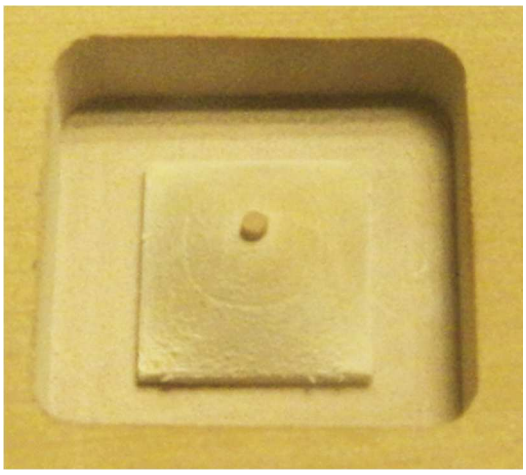


(b) Dome on top of the island

Figure 6.22 Machining a dome (wire frame display)



(a) Tool path



(b) Result

Figure 6.23 Machining a dome

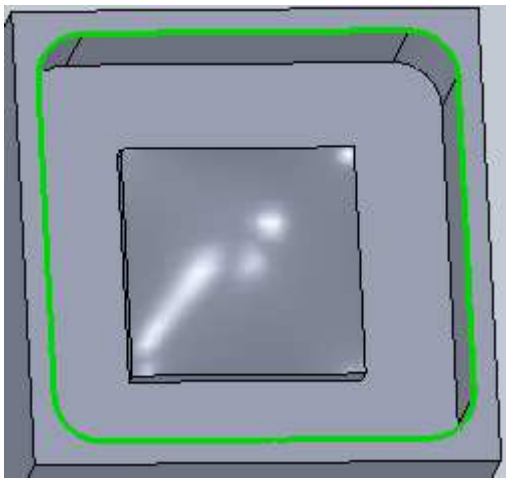
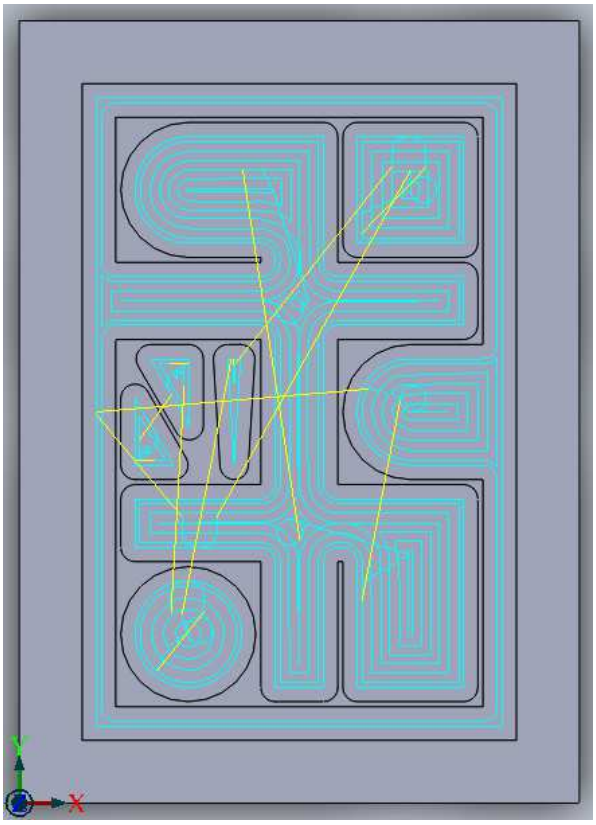


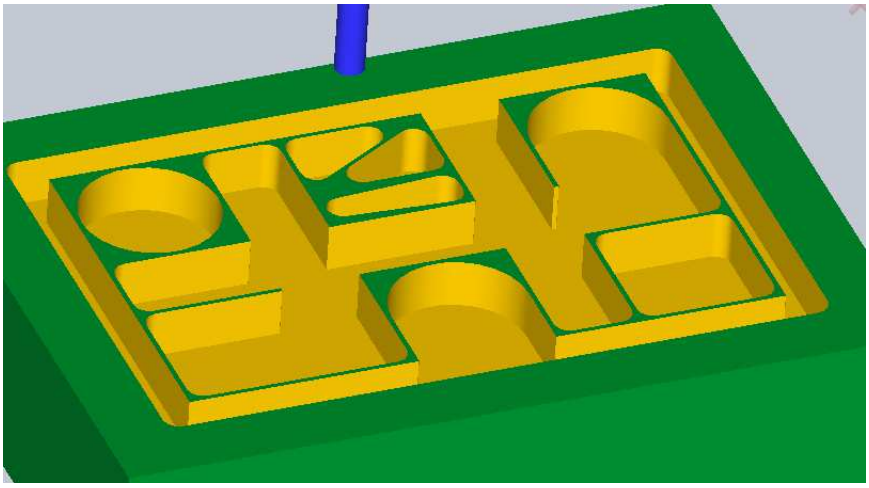
Figure 6.24 Tool containment constraint

6.4.3 Machining a Test Part

A part for testing 2.5D machining on the cooperative manipulators is designed as shown in Figure 6.25. Machining consists of several stages, in each stage a feature is machined. The depth of the cut is selected within the safe limit of the tool. Machining was done after simulation in MasterCAM. The machined work-piece is shown in Figure 6.26. It can be seen that the cooperative manipulators can machine such a part with a good finish. The cooperative manipulators was further evaluated by machining another test part with thin wall features (Figure 6.27). Thin walls with thickness of 0.1 mm were machined by the cooperative manipulators. It was observed that thin walls with lower thickness could also be machined. However, due to the brittleness of the material, the thin walls collapsed as the tool passed through them during machining. In addition, to test 3D machining, a 3D part is designed as shown in Figure 6.28. With two stages of machining, facing and finishing cut, the resulting cut can be seen in Figure 6.29.



(a) Part model and machining tool path



(b) Machining simulation

Figure 6.25 The first test part comprising of 2-axes machining features

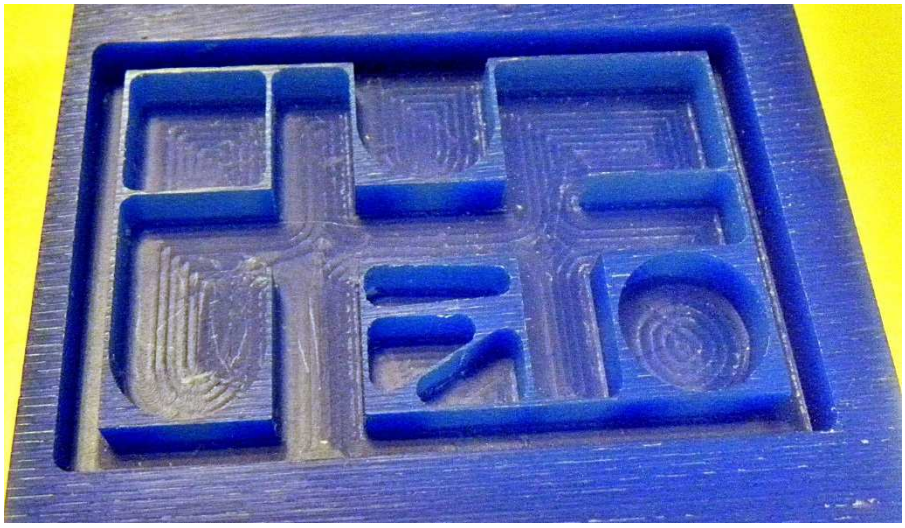
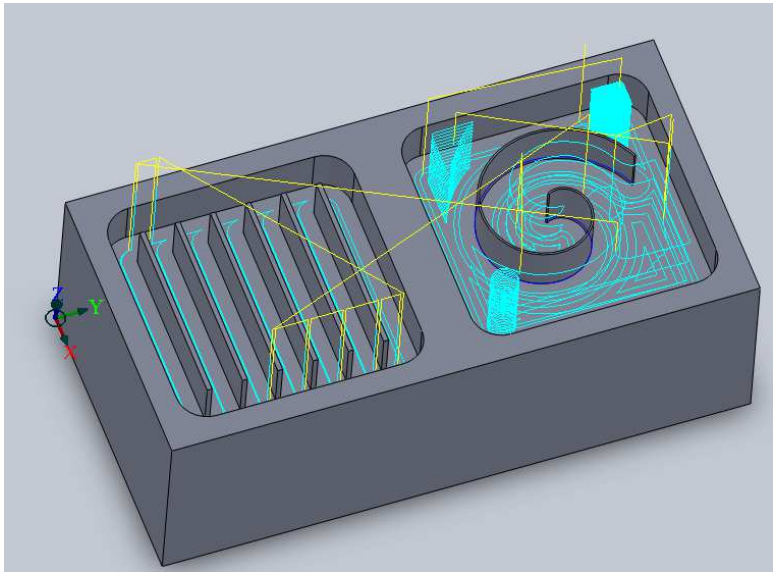
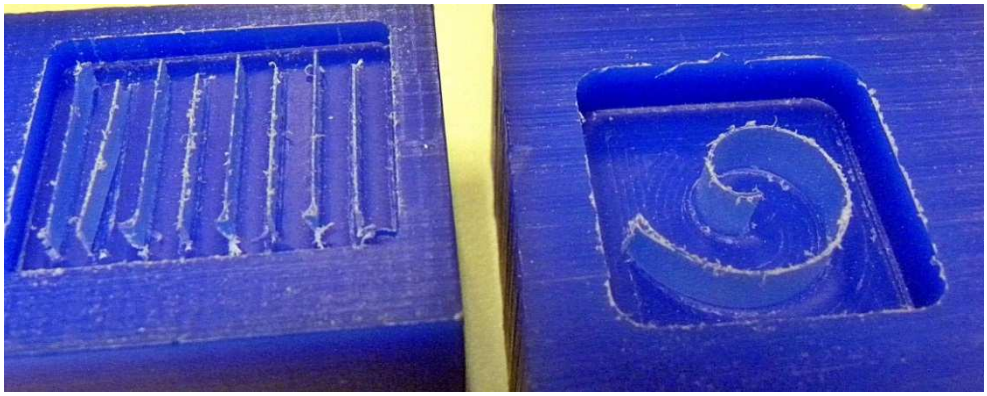


Figure 6.26 Machined test part

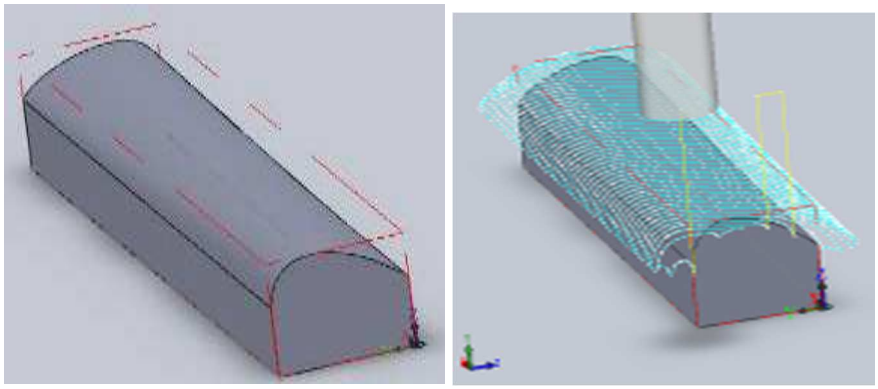


(a) Part model and machining tool path

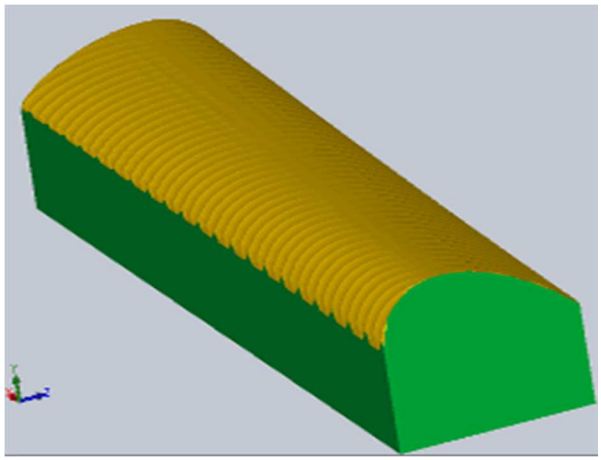


(b) Machining results

Figure 6.27 The second test part with thin wall features



(a) Part model and machining tool path



(b) Machining simulation

Figure 6.28 The part with 3-axis machining

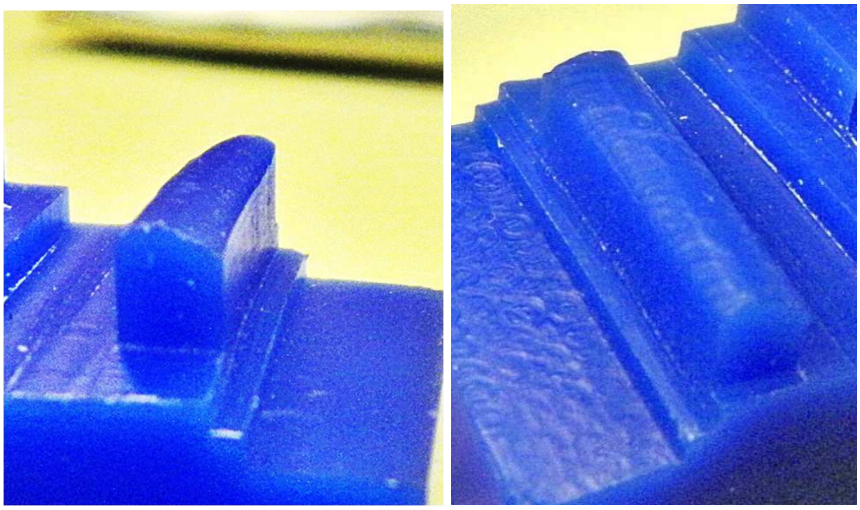


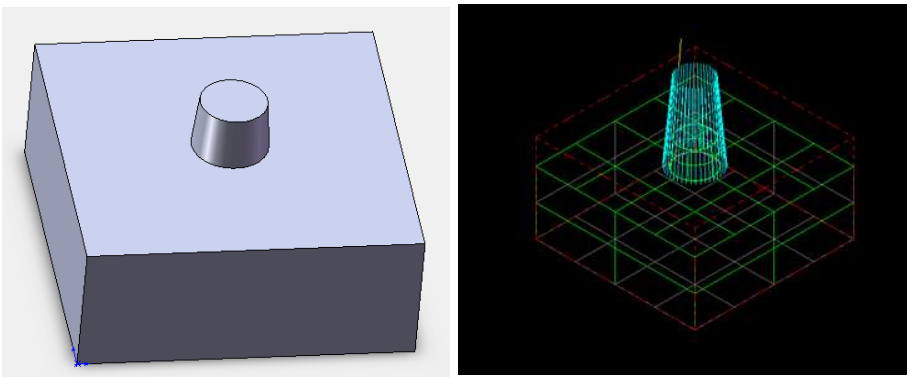
Figure 6.29 Three-axis machining result

6.4.4 Machining with Rotation Axes

Due to the limitation of the integration of SolidWorks and MasterCAM, machining with rotation axes, i.e., 4- or 5- axes machining, must be done entirely in MasterCAM. The rotation axes defined in MasterCAM must be compatible with the MatLab post-processor. Therefore, a specific tool path processor in MasterCAM is selected as the active post-processor for five-axis machining. This tool path processor has the same rotation axes and sequence as defined in the coordinate mapping for the Stewart platform.

A tapered circular island has been designed as shown in Figure 6.30. The wall of the island has an angle of five degrees. By machining this part, the Stewart platform can demonstrate the accessibility of the cutting tool to reach the tool nutation angle of five degrees while the tool precession angle changes continuously in interval [0, 360°]. The resulting work-piece is shown in Figure 6.31. In addition, an S-shape swarf wall was also designed as shown

in Figure 6.32. This shape could be a component of a more complex part such as a turbine blade.



(a) SolidWorks model (b) MasterCAM tool path

Figure 6.30 Circular Tapered Island

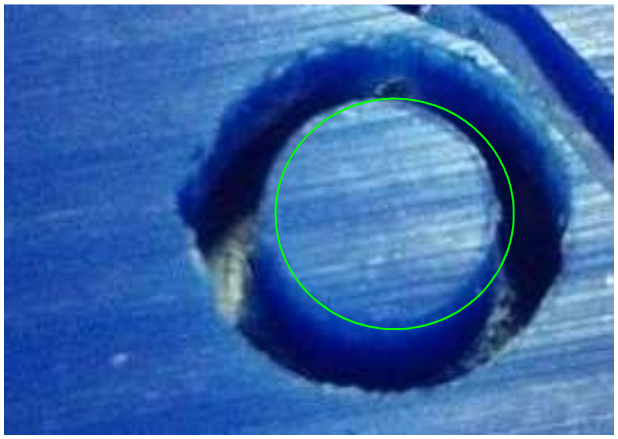
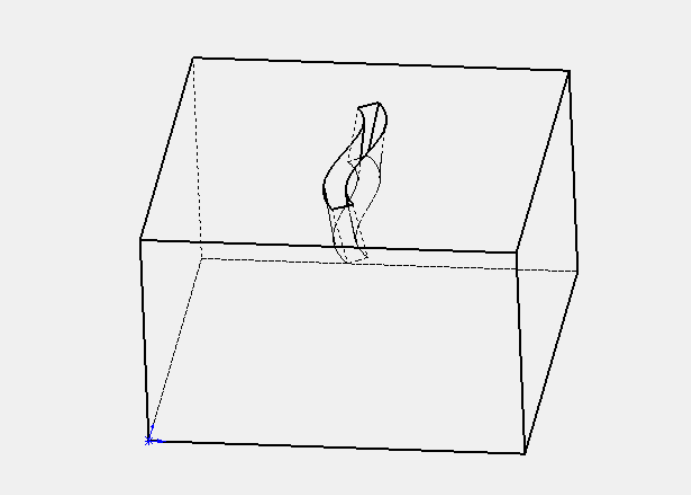


Figure 6.31 Machining result of tapered circular island on wax



(a) 3D model of the part with the S-shape swarf (wire frame display)

(b) Machining result on wax

Figure 6.32 Machining a S-shape swarf

CHAPTER 7

STEWART PLATFORM MACHINING OPTIMIZATION AND EVALUATION

7.1 Machining Workspace Analysis

The machining workspace is characterized by the locations on the work-piece that can be reached by the tool on the tool-SP. The work-piece itself is moved by the table-SP. Therefore, this is considered the farthest distant points reachable for every direction of the tool in the combined workspace evaluation. In addition, the distance between the bases of the two Stewart platforms is 1200 mm and the distance from the tool-SP moving platform origin to the tool tip is 150 mm. These data are based on the actual condition of the cooperative manipulators. To simplify the analysis, inscribed circles are used among the layers of the workspace plot as shown in Figure 7.1. This approximation guarantees that all the points in the inscribed circles are inside the actual workspace. Thus, it is named the inscribed workspace. Using the inscribed workspace, it is simplified to analyse the workspace of the cooperative manipulators.

The workspace of the cooperative manipulators is defined as the collection of tool location relative to the object resulting from the movement of the tool-SP and the table-SP. To analyse the cooperative manipulators workspace, every combination of a location in the tool-SP workspace and a

location in the table-SP workspace is tested. Consider Figure 7.2, for an instance of a 3D translation workspace combination (zero rotation). The volume of the cooperative manipulators (the combined workspace) is significantly larger compared to the volume of workspace of the tool-SP. This is due to the difference between the much bigger translational workspace of the table-SP. The table-SP can position the work-piece such that the tool-SP can reach farther location on the work-piece.

It is interesting to analyse the workspace for 4- or 5- axis machining. In order to define workspace for five-axis machining, the following description is used. While the tool tip reaches the point T_c in the workspace, the tool tilt angle can be changed in the conic face of β (Figure 7.3) . Based on this, the *smallest reachable nutation angle* (β_{min}) is defined as the smallest value of the nutation angle of the moving platform while the tool tip is at point T_c and the precession angle α changes continuously in interval [0, 360°]. The machining workspace is then defined as the collection of points that the tool tip can reach given a value of the smallest reachable nutation angle, β_{min} (Wang *et al*. 2001).

The machining workspaces of the cooperative manipulators for various tilt angles of the tool axis have been simulated. The result is presented in Table 7.1.

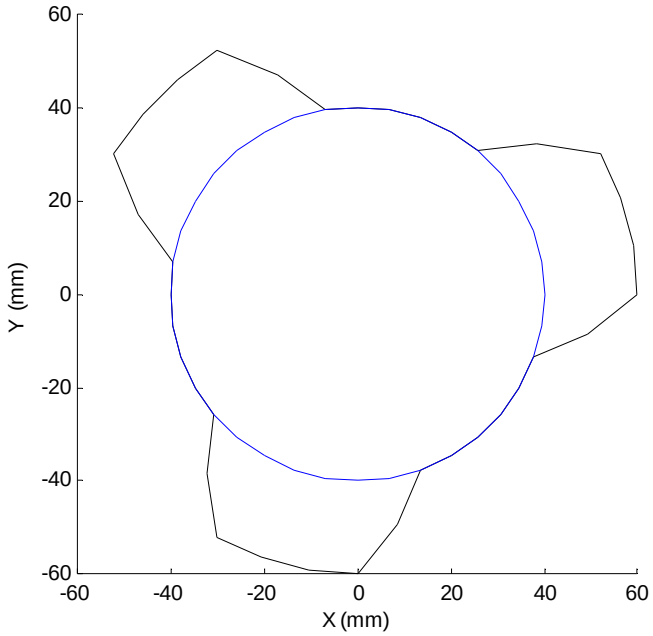
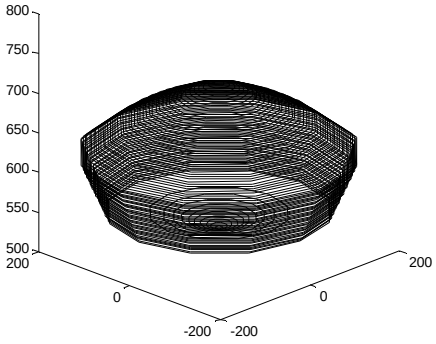
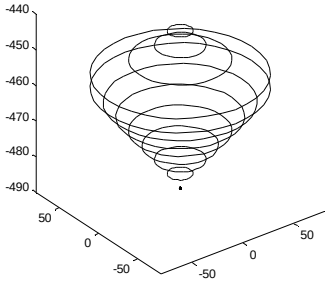


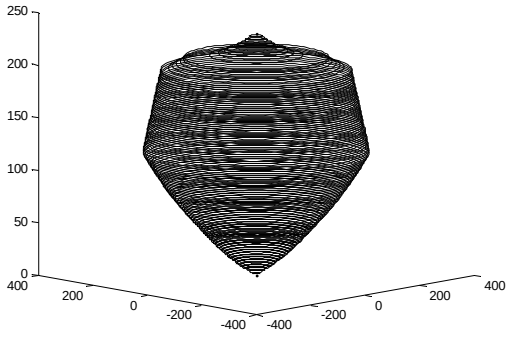
Figure 7.1 Inscribed circle (blue) inside the workspace boundary (2D workspace)



(a) Workspace of the table-SP



(b) Workspace of the tool-SP



(c) Combined Workspace of both SPs

Figure 7.2 Combination of 3D translation workspace

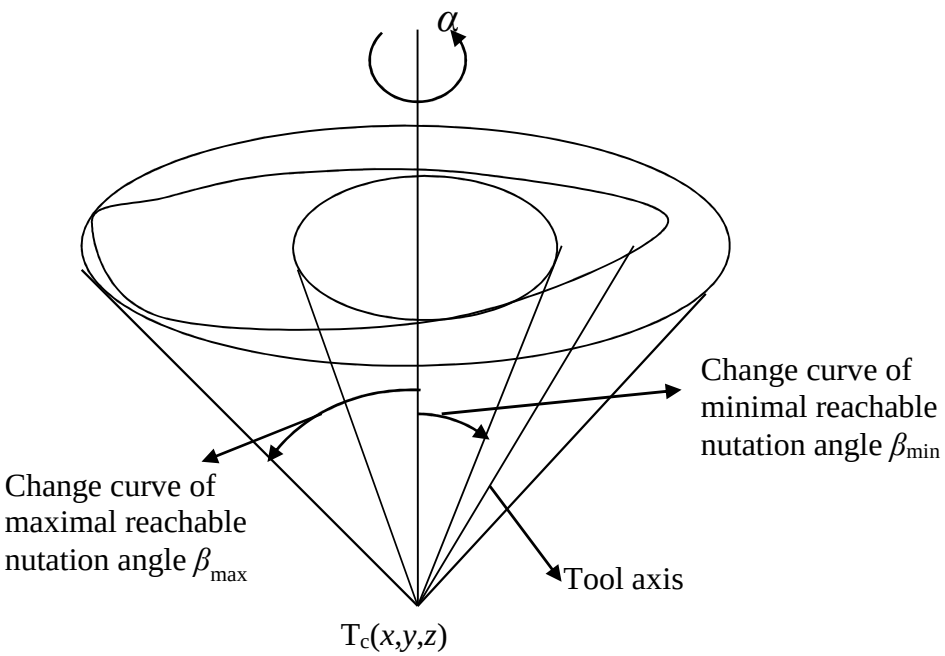


Figure 7.3 Tool axis pose in the workspace analysis

Table 7.1 Workspace Volume with various tilt angles of the tool axis

	Single Tool-SP		Extended configuration
3D Workspace ($\beta_{min} = 0$)	331,220 mm^3		33,060,000 mm^3
$\beta_{min} = 1$ degree	218,650 mm^3		31,823,000mm^3
$\beta_{min} = 2$ degree	167,140 mm^3		29,926,000mm^3
$\beta_{min} = 3$ degree	121,900mm^3		28,750,000mm^3
$\beta_{min} = 4$ degree	72,270mm^3		26,413,000mm^3
$\beta_{min} = 5$ degree	41,488mm^3		25,056,000mm^3
$\beta_{min} = 6$ degree	17,618mm^3		23,318,000mm^3
$\beta_{min} = 7$ degree	3,801mm^3		21,696,000mm^3

7.2 Application of workspace data for optimal setup in machining

For optimal machining, workspace data must be considered to utilize the full capabilities of the cooperative manipulators. The application of the information when developing a process plan and tool paths is examined. Since the workspace of the tool-SP is not a rectangle and it changes size and shape based on the tool angle, a visual method is called for. The method is to use the 3D plot of the tool-SP workspace for a given tool angle in the CAD/CAM system to choose the location and orientation of the work-piece. The work-piece is stationary in the single configuration and it is fixed on the machining table. In the extended configuration, the work-piece is carried by the moving platform of the table-SP. Therefore, the work-piece location and orientation can change during machining which allows the tool-SP to access a larger volume.

Several parameters influence the work volume as it is applied to process planning and tool path generation. First, the volume is based on the position of the moving platform, and the orientation of the tool axis vector. If the tool is not tilted with respect to the table, three-axis machining, the workspace takes on a different shape, in this case larger. Large tool axis vector tilt angles would be required to remove the needed material, if the volume is much smaller. This is the case because positioning uses a finite resource, the available lengths of the legs.

Because only five DOF motions are used for machining, the rest of the DOFs can be used for optimizing certain indices. However, to minimize error in the motion and maintain accuracy in machining, it is advised that only the tool-SP is used to cut the material and the table-SP is stationary in processing a trajectory. The focus of the table-SP is to provide larger workspace for the tool-SP if required. Therefore, the table-SP is used to position and orientate the object or work-piece with respect to the tool-SP and the tool-SP is used to perform the intended operation or the cutting of the material.

7.3 Calibration and Accuracy Improvement

Manufacturing tolerances of building a manipulator often cause an inherent problem where the kinematic parameters are not exactly equal to the values in the kinematic model. Kinematic calibration is a process of identifying the actual values of the kinematic parameters in the kinematic model. Thus, by updating these parameters, the inverse kinematic calculation

of the required joint angles will result in an accurate end-effector pose. The calibration process generally consists of four basic steps, namely (1) development of a kinematic model that contains a set of parameters to determine the relationship between the actuated joint angles and the end-effector pose, (2) measurement and recording of the manipulator poses, (3) error minimization through searching for the optimum kinematic model parameters of the manipulator from the pose measurements and manipulator actuated joint angles, and (4) correction for the geometric parameter errors in the manipulator kinematic model.

The kinematics formulas are crucial in the control of a PKM to move its end-effector to a desired location. However, this requires the PKM to be built according to the nominal kinematic parameters, such as joint locations and leg offsets. The main purpose of the calibration is to find the actual kinematic parameters that have deviated from their nominal values due to the imperfect assembly and manufacturing tolerances.

The first step of calibration is error modelling. Error modelling is important for profiling geometric entities that cause motion errors of the end-effector. Error modelling is the process of determining the parameters that affect the motion of the end-effector. Errors considered in the calibration are geometric errors and they are treated as static values or constants. The following explanation uses the designation for the tool-SP, although the same derivation can be made for the table-SP.

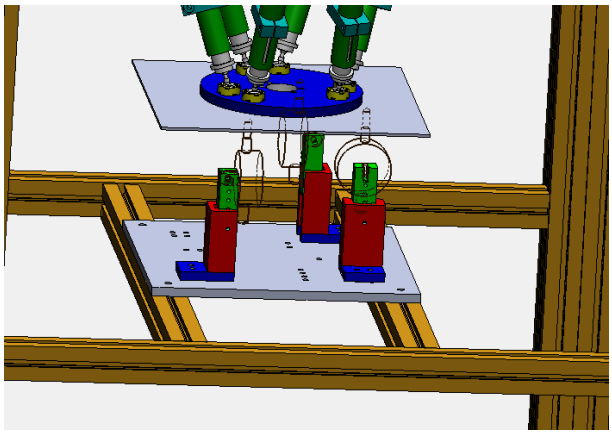
It is generally accepted that 42 kinematic parameters can fully describe a configuration of a nominal Stewart platform (Wang and Masory

1995). These kinematic parameters are of main importance and they are namely, the leg length offset L_{Oi} (one parameter), locations of the universal joints, b_i (three parameters each), and locations of the spherical joints, a_i (three parameters each). Hence, there are seven kinematic parameters per leg.

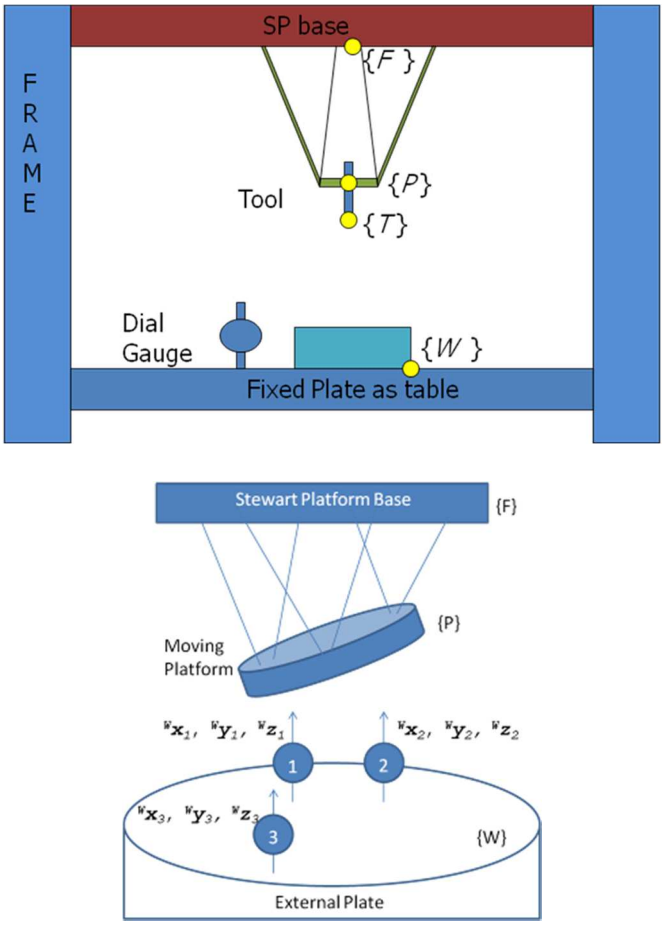
Kinematic calibration can solve kinematic parameters with respect to an arbitrarily fixed coordinate system located at the base frame $\{F\}$ and the moving platform $\{P\}$. The absolute locations of the origins of frame $\{F\}$ and frame $\{P\}$ have no effect on the motion error, because the kinematic parameter errors are defined within the Stewart platform closed kinematics loops relative to these origins. Therefore, six parameters that are defined with respect to frame $\{F\}$ and six parameters with respect to frame $\{P\}$ can be eliminated or excluded from the calibration. In the calibration, the parameters b_1, b_2, ${}^{P}a_1$, ${}^{P}a_2$ are set to the nominal/reference values and the other 30 parameters define the kinematic configuration of the Stewart platform.

Measurement data are taken from the digital indicators during calibration. These digital indicators provide additional redundant sensing which is used to calibrate the Stewart platform. The digital indicators are installed on the machining table (single configuration) with reference to the world coordinate system. There is no specific constraint on the position of the digital indicators; however, they must be installed with the axes of the gauges perpendicular to the table. The purpose is to measure the distance along the axes of the gauges between the machining table and the moving platform. Figure 7.4 shows an illustration of the Stewart platform with the setup. The coordinate frames $\{T\}$ and $\{W\}$, refer to the tool coordinate system and the

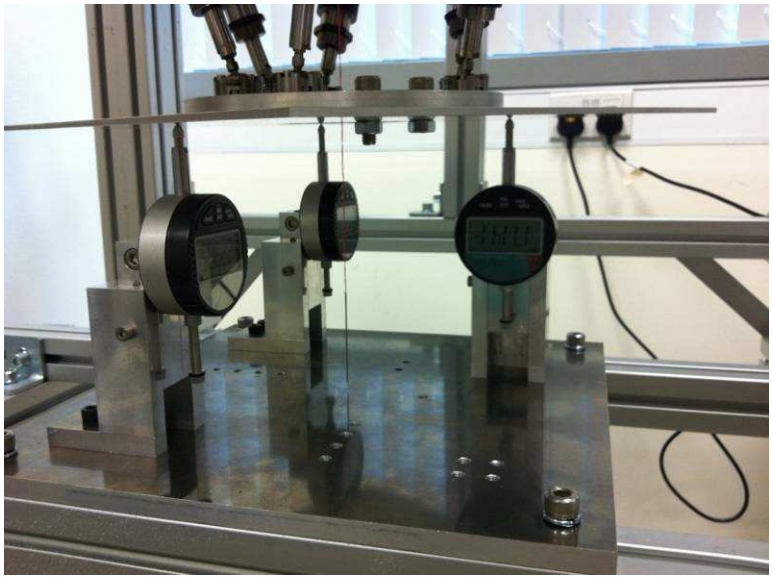
world coordinate system, respectively. The digital indicator measurement is achieved relative to the world coordinate system. The world coordinate frame, $\{W\}$ is attached on the machining table on which the digital indicators are installed, as the reference for the measurement.



(a) 3D model



(b) Coordinate systems



(c) Dial gauges for measurement

Figure 7.4 Calibration setup

A numerical nonlinear gradient descent optimization method is used to solve for the real kinematic parameters from the measurement data. The digital indicators are measurement devices to verify the location of the moving platform and determine the error between the desired and the actual locations. The readings of the digital indicators give different values than the expected values (nominal values) if there is a discrepancy or an error in the moving platform location. An error at one pose can be formulated as $Y_i - F_i(Q_i, \eta)$, i=1..N, where η denotes the real (unknown and to be found) kinematic parameters, Y_i is the measurement vector taken from the readings of the indicators, and Q_i is the leg length computed from the inverse kinematics corresponding to the i^{th} pose when the measurement is taken. F_i is a function that gives the nominal values of readings of the digital indicators when the moving platform is at the desired location. When there is no error Y_i is equal to F_i. In other words, if there is no difference in the nominal kinematic parameters and the real kinematic parameters, there would be no kinematic errors. The calibration is solved by optimization that is formulated as Equation (7.1).

$$\min_{\eta} \left\| \begin{matrix} Y_1 - F_1(Q_i, \eta) \\ \dots \\ Y_N - F_N(Q_i, \eta) \end{matrix} \right\| \tag{7.1}$$

N is the number of measurement poses taken. Equation 7.1 will minimize the error by finding the real kinematic parameters. This equation is also known as the Least Squares Formulation (LSF). The LSF procedure

determines the real kinematic parameters, which are initially set according to the nominal parameters of the Stewart platform. When the errors in Equation 7.1 are minimized, the set of kinematic parameters that best represent the actual kinematic parameters of the Stewart platform can be found.

The forward kinematics calculation is performed in each function, F_i to obtain the expected values of readings of the digital indicators. The function F_i in Equation (7.1) gives the pose of the moving platform from the leg lengths, Q_i and a set of kinematic parameters, η. The function F_i computes the expected digital indicator readings based on the machining table location, on which the frame $\{W\}$ is attached, with respect to the base coordinate frame $\{F\}$. Using the end-effector location that has been calculated, the expected digital indicator readings are determined by finding the distances between the machining table and the moving platform.

Simulation is carried out to verify the calibration method before applying the method to the actual Stewart platform calibration with real data. In the simulation, a real model is assumed and used to test the calibration method. Table 7.2 summarizes the deviations of the real model from the nominal model. The nominal model is used for initialization of the kinematic parameters at the start of the error minimization in the LSF. Table 7.3 presents the various simulation results with a random error applied in the measurement data. This error emulates the noise and disturbance in the measurement. The error is modelled as the Gaussian white noise with the corresponding variances (noise level) listed in Table 7.3.

The calibration process has been performed in several stages to improve and obtain the best geometric parameters that can minimize the errors. Some measurement errors due to instability were observed during the calibration process where the data obtained for the same pose differed by around 10-50 micrometre, depending on the previous state of the Stewart platform. This could be caused by the backlash in the passive joints. To compensate this error, two sets of data were taken for each configuration during measurement and the average was recorded. The result shows the calibration using optimally selected poses gives the least error. Table 7.4 summarizes the calibration result with a set of 110 poses.

Note that each SP is calibrated with respect to the local coordinate frame which origin is at the SP's base. While it is true that the calibration of each SP may yield worse result than the simultaneous calibration of both SPs, the latter is actually not feasible for practical reasons. For the calibration to work, the coordinate frame must be fixed accurately. In the setup of 2-SP inside a custom-made frame, the relative location of both platforms with respect to each other is not known beforehand. Although the values can be approximated, a small error will lead to wrong geometric parameters. In addition, for simultaneous calibration over 70-80 parameters will need to be optimized. The number of parameters is quite large and the optimization has low chance to find good representative geometric parameters, especially if approximation is used.

Table 7.2 Real model for calibration simulation

Assumed leg lengths offset error (mm)					
δL_{O1}	δL_{O2}	δL_{O3}	δL_{O4}	δL_{O5}	δL_{O6}
1	2	3	1	2	3
Assumed position error of the base joints (universal joints) (mm)					
	δb_1	δb_2	δb_3	δb_4	δb_5	δb_6
X	0	3	0.1	-0.1	-0.1	-0.1
Y	0	0	1.8	1.8	1.8	1.8
Z	0	0.3	0.7	0.7	0.7	0.7
Assumed position error of the platform joints (spherical joints) (mm)					
	δa_1	δa_2	δa_3	δa_4	δa_5	δa_6
X	0	-0.9	3	3	3	3
Y	0	0.67	-2	-2	-2	-2
Z	0	0.53	0.7	0.7	0.7	0.7

Table 7.3 Error comparison of calibration simulation

NL	EE	DE	PEb	Pea	Per
0.0001	99.99%	99.81%	0.12, 3.11, 1.27	0.0006 ,0.001 ,0.002	99.41%, 99.97%, 99.85%
0.001	99.95%	99.57%	0.12, 2.99, 1.26	0.002, 0.004, 0.003	98.28%, 99.87%, 99.79%
0.01	99.48%	97.78%	0.12, 2.78, 1.24	0.01, 0.006, 0.023	90.90%, 99.78%, 98.13%
0.1	93.80%	94.61%	0.12, 3.24, 1.23	0.014, 0.05, 0.04	88.15%, 98.59%, 96.60%

Column Info: NL: Noise Level (variance) , EE: reduction in estimation error (sum of squares of $F_1..F_N$), DE: reduction in mean difference between nominal and real digital indicators reading , PE(x): pose error mean in α (degree), β (degree), and ${}^{F}z_0$ (mm), respectively (PEb:before, PEa:after, PEr: error reduction).

Table 7.4 Kinematic Parameters after Calibration

Leg lengths offset (mm)					
L_{O1}	L_{O2}	L_{O3}	L_{O4}	L_{O5}	L_{O6}
1	2	3	1	2	3

Position of the base joints (universal joints) (mm)						
	b_1	b_2	b_3	b_4	b_5	b_6
X	-152.5	-129.069	76.15	131.9689	76.15	-0.1
Y	0.0	76.25	130.269	-74.45	133.869	154.3
Z	0.0	0.3	0.7	0.7	0.7	0.7

Position error of the platform joints (spherical joints) (mm)						
	${}^{P}a_1$	${}^{P}a_2$	${}^{P}a_3$	${}^{P}a_4$	${}^{P}a_5$	${}^{P}a_6$
X	-60.622	-0.9	3	73	63.22	-32
Y	35	0.67	-72	-2	33	58.622
Z	0.0	0.53	0.7	0.7	0.7	0.7

7.3.1 Perpendicularity of Dial Gauges

In the case that the dial gauges are not perfectly mounted onto the external plate, additional parameters representing the error angle on the perpendicularity for each gauge, Υ_i, i=1..3. This error angle can be described in Figure 7.5. The dial gauge measurement axes are supposed to be parallel to Z_W. However, due to assembly error and manufacturing tolerance, it may have deviation. Thus, the actual distance reading from each gauge should be corrected by a factor of $cos\ \Upsilon_i$ for each dial gauge. These parameters can be included in the optimization to solve the problem as given in the following equation.

$$F_i(Q^0 + \Delta Q, Y^0 + \Delta Y, \eta^0 + \Delta\eta),\ i=1..N \quad (7.2)$$

where the superscripts 0 indicate that the variables or parameters are at their nominal or supposed values (measured or selected) and Υ_i is included in the measurement error $\Delta Y_i = Y_i - Y_i.\cos(\Upsilon_i)$. An extended Kalman Filter or also known as the Total Least Squares Formulation (TLSF) method is proposed to

counter this problem. *A priori* notions of accuracy of the measurement that can be modeled using statistical approach with a maximum-likelihood principle are considered in the TLSF. In this context, the LSF does not take into account measurement and input errors, $\Delta Q = 0$ and $\Delta Y = 0$. In other words, it is assumed that the measurement is perfect and the real leg lengths are known without any discrepancies. Consequently, it only finds the $\Delta\eta$ that minimizes the discrepancies.

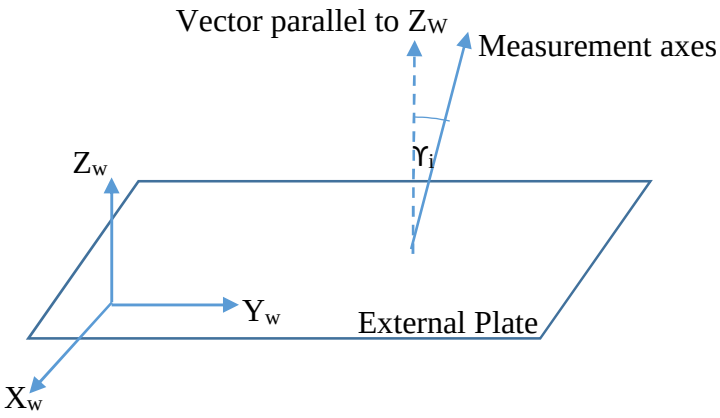


Figure 7.5 Error angle on dial gauges

On the other hand, the TLSF needs to consider non-zero ΔQ and ΔY. If this is the case, the TLSF must find the best combination of ΔQ, ΔY, and $\Delta\eta$. Because these errors are assumed to be Gaussian with zero mean, the best combination is the minimizer of (Patel and Ehmann 2000; Wampler *et al.* 1995):

$$\chi^2 = \sum_{i=1}^{N} \Delta Q^T W_Q \Delta Q + \Delta Y^T W_Y \Delta Y + \Delta\eta^T W_\eta \Delta\eta \qquad (7.3)$$

where W_Q, W_Y, W_η can be obtained from the inverse of the covariance matrices for ΔQ, ΔY , and $\Delta\eta$, respectively. Thus, TLSF minimizes (7.3) subject to (7.1). With the TLSF, it is ready to calibrate the real case with imperfect information from measurements. However, prior to real calibration, TLSF simulations are performed with the same settings with the previous simulation using the LSF. Equation (14) shows that the TLSF tries to guess what is happening in the real case where errors are not just caused as a result from only the kinematic parameters deviation, but also error angle, noises and other un-modeled errors in terms of joints coordinates. Table 7.5 summarizes the calibration results with error angles imposed on the dial gauges.

Table 7.5 Error comparison of calibration with measurement errors

NL	EE	DE	PEb	PEa	PEr
0 (LSF)	99.92%	97. 703%	1.09, 0.61, 1.26	0.021, 0.028, 0.018	98.09%, 95.31%, 98.6%
0.0001	92.67%	66.76%	2.96, 2.99, 2.92	1.98, 2.93, 2.18	32.88%, 2.12%, 25.48%
0.001	99.52%	71.53%	2.54, 2.81, 3.15	2.06, 2.62, 1.97	18.92%, 6.99%, 37.54%
0.01	99.05%	67.02%	2.28, 2.73, 3.38	2.11, 2.78, 1.84	7.29%, -1.73%, 45.61%
0.1	50.25%	26.64%	1.903, 2.79, 1.82	1.95, 2.76, 1.65	-2.51%, 1.03%, 8.82%

Column Info: NL: Noise Level (variance) , EE: reduction in estimation error (sum of squares of $F_1..F_N$), DE: reduction in mean difference between nominal and real digital indicators reading , PE(x): pose error mean in α (degree), β (degree), and ${}^{F}z_0$ (mm), respectively (PEb:before, PEa:after, PEr: error reduction).

7.3.2 Pose selection for Calibration

When the calibration is being performed, measurement data are collected from the various poses or configurations of the PKM. These data, however, can be affected by noise and physical instability. The set of measurement poses can be optimized to yield robust calibration. The quality of a certain set of configurations/poses with respect to the calibration can be approximated based on an observability index that can be obtained from the identification Jacobian matrix, $\boldsymbol{J}_P$. This matrix has components that are calculated using the linearized version of Equation (7.1)

$$\Delta Y = \boldsymbol{J}_P.\left(\boldsymbol{X}_P\right)\Delta\eta \tag{7.4}$$

In Equation (7.4), ΔY is the difference in the measurement variable (digital indicator readings) when an error $\Delta\eta$ is induced in the kinematic parameters, and $\boldsymbol{X}_P$ is the pose set (collection of the moving platform location where measurements are taken) used in the calibration. The identification Jacobian matrix, $\boldsymbol{J}_P$, can be computed numerically by assuming small difference in $\Delta\eta$ (Huang *et al*. 1998). There are four options that have been proposed in the literature for the definition of the observability index from the matrix $\boldsymbol{J}_P$ (Nahvi and Hollerbach 1996; Daney 2002). One of the best indices that relate well to error calibration sensitivity is the Noise Amplification Index, O given in equation (5), where σ_i are the singular values of $\boldsymbol{J}_P$ ordered from largest to smallest so that $\sigma_1 >= \sigma_n$.

$$O = \sigma_n^2 / \sigma_1. \tag{7.5}$$

In this research, a swarm-based PSO (Particle Swam Optimization) (Kennedy and Eberhart 1995) search method shown in Figure 7.6 is proposed to search for the best set of poses for calibration measurement. The PSO method models the optimum candidates as moving particles over the search domain, which is the PKM workspace. The aim is to find a pose set within the workspace of the PKM that maximizes the objective function $O(\boldsymbol{J}_P(\boldsymbol{X}_P))$, where O() is a function to compute the observability index of $\boldsymbol{J}_P(\boldsymbol{X}_P)$, which is the identification Jacobian matrix characterizing the error relationship between the kinematic errors and the measurement variables. The framework of this algorithm is a modification of the one proposed in (Daney *et al*. 2005), but with PSO as its search method. The steps are as follows:

1. Initial set, $X_1 \ldots X_N$ for N poses needed for selection, is formed by random poses in the workspace.
2. AddFind step: Using PSO, find an additional X_{N+1}, that maximizes $O(\boldsymbol{J}_P(\boldsymbol{X}_P))$ where $\boldsymbol{X}_P$ includes $X_1 \ldots X_N$, and X_{N+1}.

a. To find X_{N+1}, randomize the particles over an allowable range of the search domain. Each particle represents a solution candidate for X_{N+1}.

b. Compute the best position of each particle and find the global best particle position (the best candidate that gives the highest objective value among all the particles). The objective value is computed from $O(\boldsymbol{J}_P(\boldsymbol{X}_P))$ where X includes $X_1 \ldots X_N$, and the pose represented by the position of each particle being evaluated.

c. Update the particle position (move the solution candidates to new positions) based on standard PSO governing equation using the information from the previous step.

d. Stop if the global best objective value is not improving for several iterations or the number of iterations exceeds a predefined limit. The X_{N+1} found is the pose represented by the last global best particle position.

e. RemoveFind step: Find an X^- within the set $X_1 \ldots X_{N+1}$ that if this X^- is removed from the set, the remaining poses give a higher objective value of $O(\boldsymbol{J}_P(\boldsymbol{X}))$ where $\boldsymbol{X}$ includes all poses in $X_1 \ldots X_{N+1}$ except X^-.

4. Repeat steps (2) and (3) until $X^- = X_{N+1,}$ which implies that the newly added pose is not improving the current pose set, and stop.

The initial pose set is initialized randomly over the workspace. Next, the PSO is conducted to search for one additional pose such that if this pose is added to the group, the group will have the best performance index; this is the AddFind step. Immediately after a new pose has been added to the group, the RemoveFind step is executed to find one pose of the group that if this pose is removed, the group will have the best objective value. These two main processes, AddFind and RemoveFind, are repeated until no better objective value for the pose set can be found. The PSO search domain is defined by the boundary of the Stewart Platform PKM workspace. In addition, the optimum configurations are often found near the boundary of the workspace. This

implies that the initial guesses of the PSO search can be taken near the boundary to expedite the search process.

Figure 7.7 shows the global best performance index (observability) optimized in the AddFind process based on the global best pose. The PSO algorithm is stopped after 40 iterations and it is concluded the pose with the global best performance as the candidate to be added to the pose group in AddFind. RemoveFind is implemented sequentially to check each and every pose in the group by calculating the index. If one pose is added, and then one pose is removed in the AddFind-RemoveFind cycle. If the pose to be added and the pose to be removed are the same, then the search for pose stops.

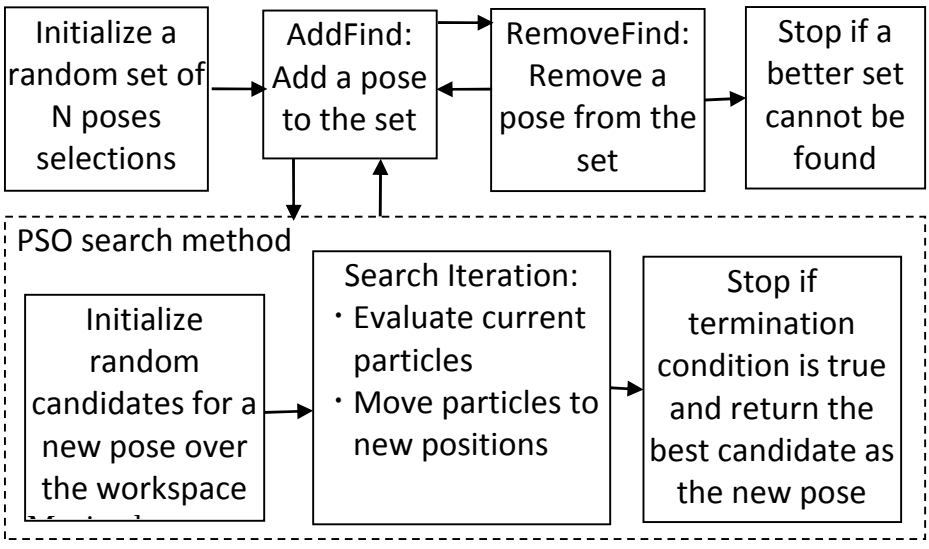


Figure 7.6 PSO algorithm for optimum poses selection

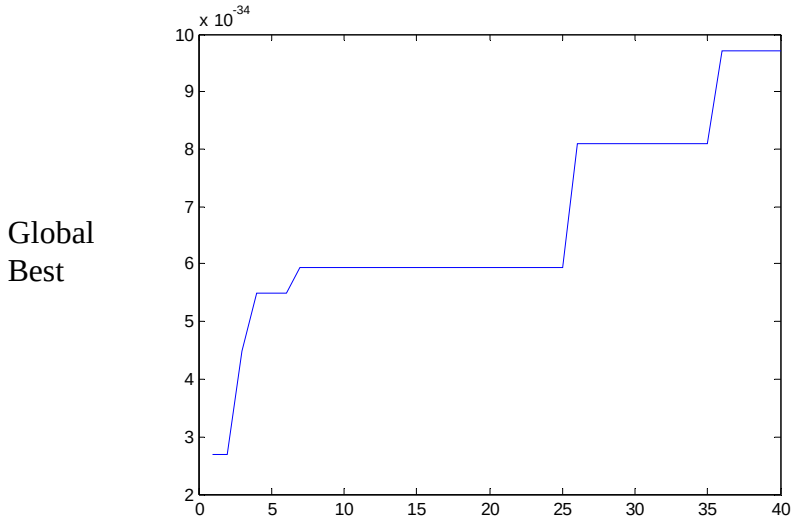


Figure 7.7 Global best evaluation in PSO search

The number of poses required to obtain a good calibration result has been investigated. Figure 7.8 presents the plot of the error in the kinematic parameters after calibration against the number of pose measurements. In this test, random poses are used. It can be concluded that a larger number of poses will lead to better calibration results. In addition, there is a limit where the errors cannot be reduced further with additional number of poses.

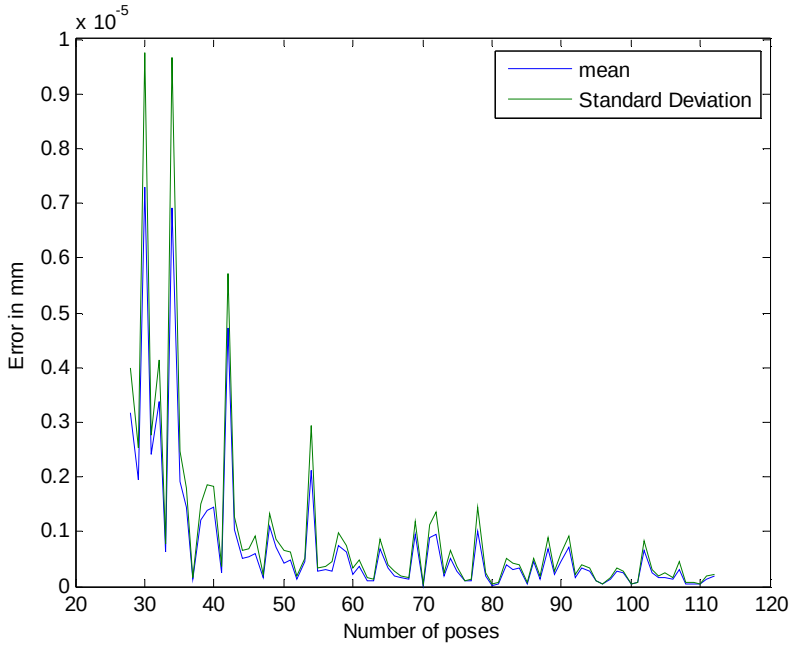


Figure 7.8 Calibration improvement with a larger number of poses

7.3.3 Online Calibration for Kinematic Parameters Error Compensation

The proposed calibration has potential for automated calibration as long as the selected poses are measurable and reachable. The calibration procedure presented in the previous section is extended to form a self-calibration procedure to improve the accuracy of the PKM when it is in operation. The measurement method for the self-calibration procedure is using the draw-wire sensors installed between the base and the moving platform of the Stewart platform. The objective is to form a passive 6-legged mechanism using the wire sensors, so that the wire sensor readings can fully capture the moving platform pose. As the wire sensors are used for the calibration procedure, there is no interference between the sensors and the

Stewart platform. This arrangement allows machining and measurement to be performed at the same time (Figure 7.9).

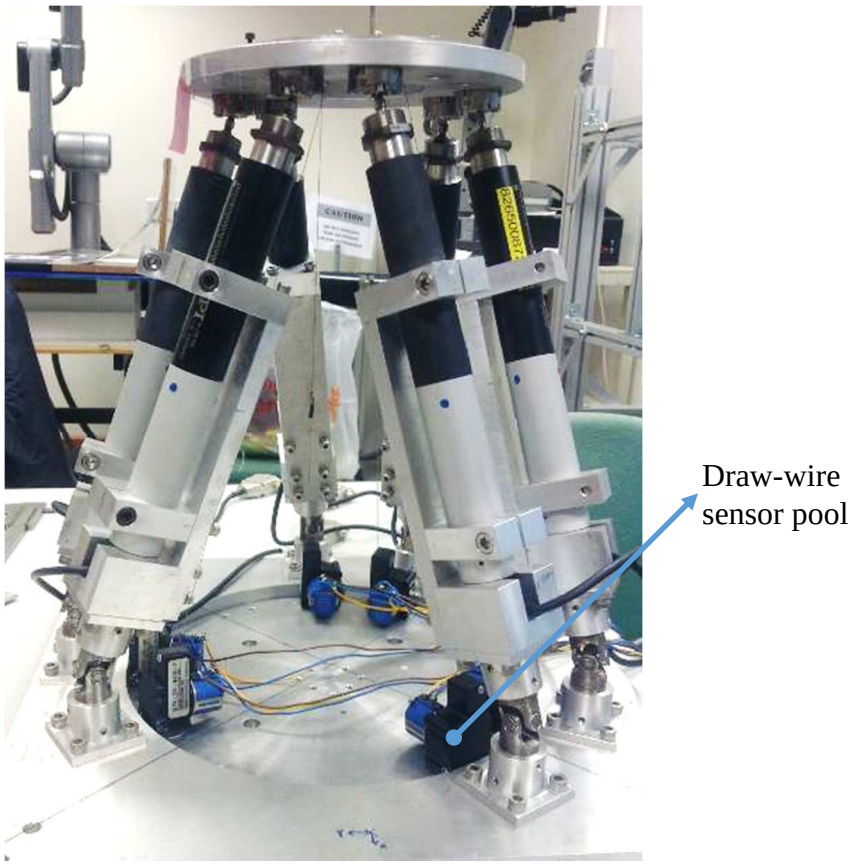


Figure 7.9 Draw-Wire Sensor Assembled to tool-SP

Under a constant voltage input, the draw-wire sensor will have different output according to the length of the wire that has been drawn out of the sensor. The output is a voltage signal instead of the reading of the wire length. Hence, a calibration of the draw-wire sensor is also needed to find the relationship between voltage output and the wire length as well as to check the linearity of the output.

In order to stabilize the voltage input to have a steady result, an external power supply is used. A height gauge with an accuracy resolution of 10 μm is used to calibrate the wire sensor. The results shown in Figure 7.10 show good linearity. With these data, the bias offset in voltage reading is obtained. With interpolation or extrapolation, the length of the drawn wire can be calculated.

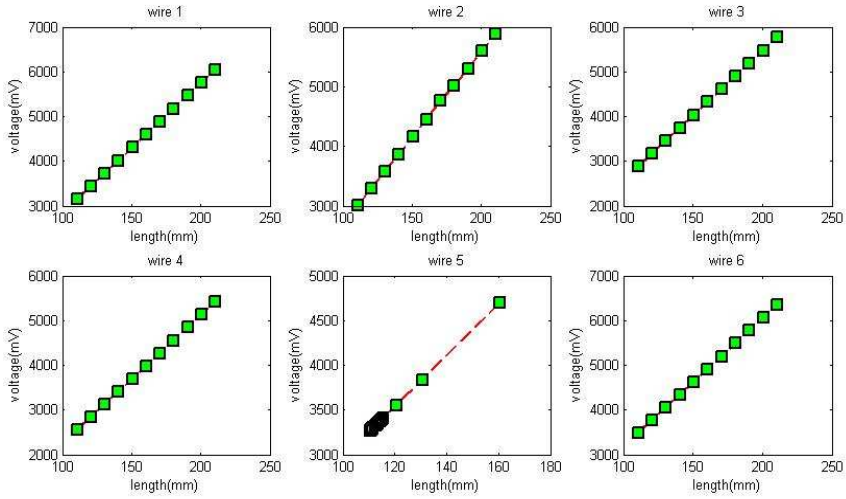
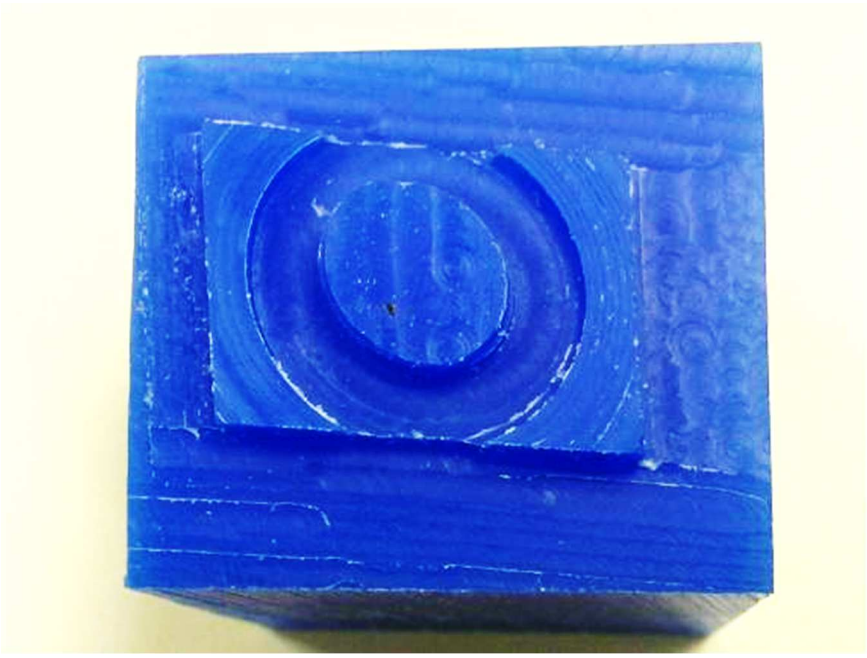


Figure 7.10 Relationship between voltage output and Draw-Wire sensor length

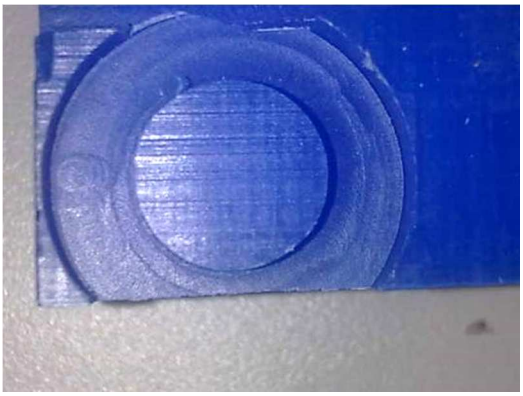
7.4 Machining Evaluation

A circular pocket was created using the cooperative manipulators before and after the calibration. Early investigation shows that the accuracy of the Stewart platform differs significantly such that it can be seen in the shape of the work-piece in Figure 7.11. Before calibration, the shape is more like an ellipse than a circle. The accuracy of Stewart platform improves after

calibration took place. The updated kinematic parameters resulted in a better shape for the created part.



(a) before calibration



(b) after calibration

Figure 7.11 A circular pocket part during calibration

Further, facing operation was performed to observe the smoothness of the surface such as shown in Figure 7.12. With careful selection of the

distance between two subsequent via points, the interpolation can give sufficient number of points in order for the tool-SP to machine a good surface finish.

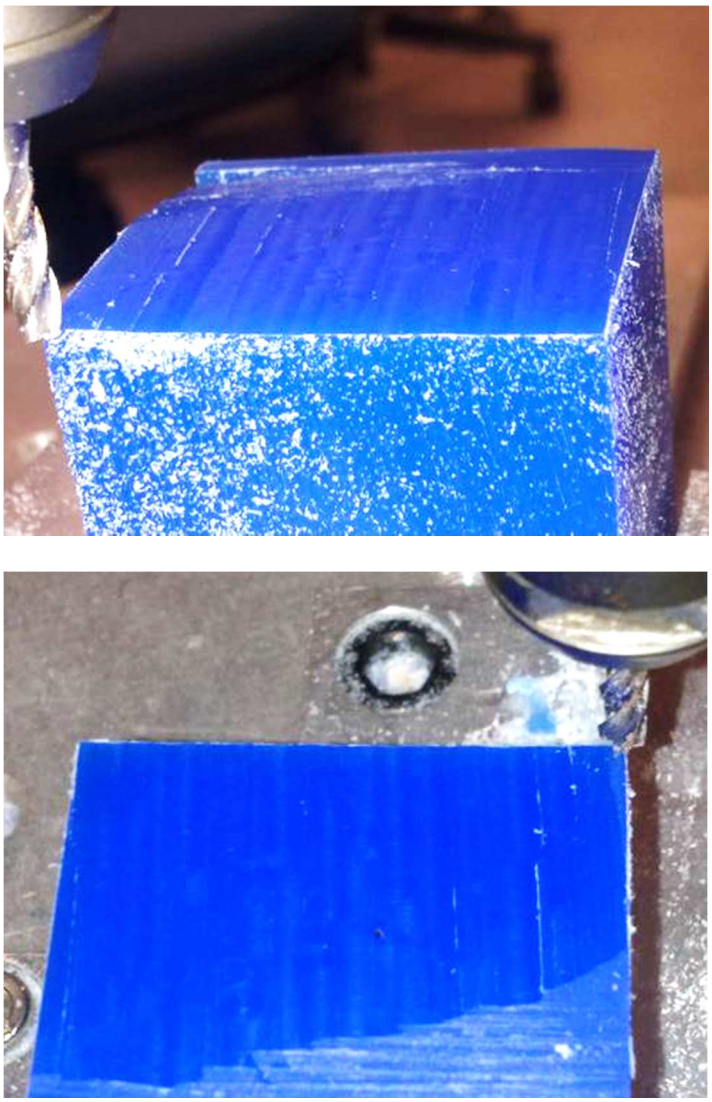


Figure 7.12 Facing operation of a work-piece

In addition, although machining can be performed using the cooperative manipulators without any problems, the chips accumulated from the work-piece could not be removed automatically. In this cooperative

manipulators, the author tried his best to clear off the chips during machining (Figure 7.13). The material of the work-piece is relatively soft compared to metal. Hence, no tool jam was observed although the chips were piling around the work-piece during machining. However, for metal cutting the chips could jam the tool. An automated chips removal system may need to be installed on the cooperative manipulators for metal cutting. In this research, the study is limited to machining wax because it is soft and can be machined quite easily. The machined part can reflect the tool movement of the cooperative manipulators and can thus ensure safe operation.

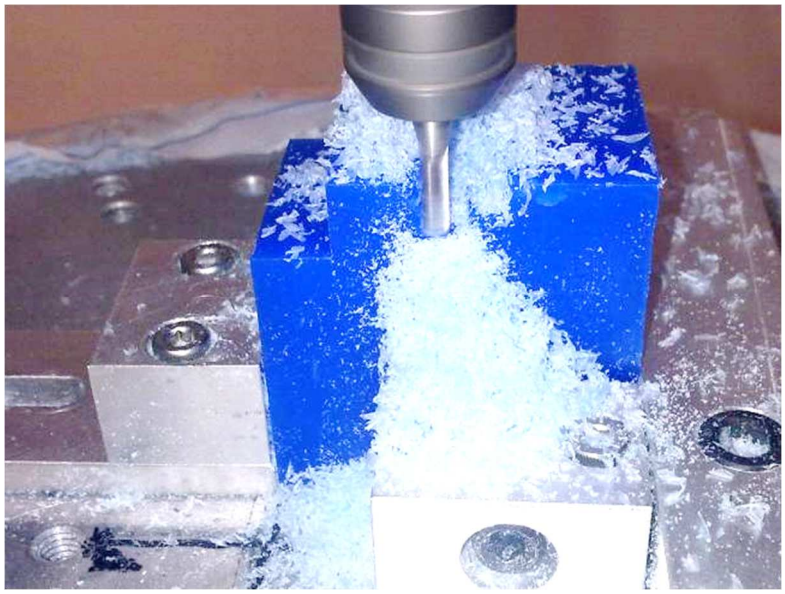


Figure 7.13 Chip accumulation during machining

In order to test the rotation capability of the tool-SP (in single configuration), a part with a tapered wall was machined. The inclination angle of the wall is 10 degrees. The maximum angle in one axis of rotation of the tool-SP is about 13-15 degrees. The result is shown in Figure 7.14. In addition,

Figure 7.15 shows the tool-SP when it was machining the tapered wall. As can be seen, at the configuration during machining the tapered wall, the tool-SP has almost reached the limit of some of its ball-socket joints. In fact, the ball-socket joint constraint is the main limitation for the rotation of the tool. Thus, the table-SP in the extended configuration could be used to position the work-piece to allow the tool to approach the work-piece with a larger rotation angle.

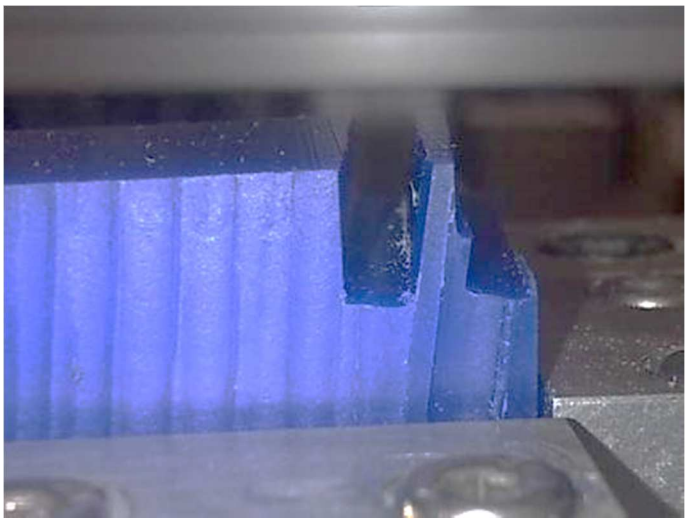


Figure 7.14 A tapered wall machined from the most tilted angle of the tool of the tool-SP

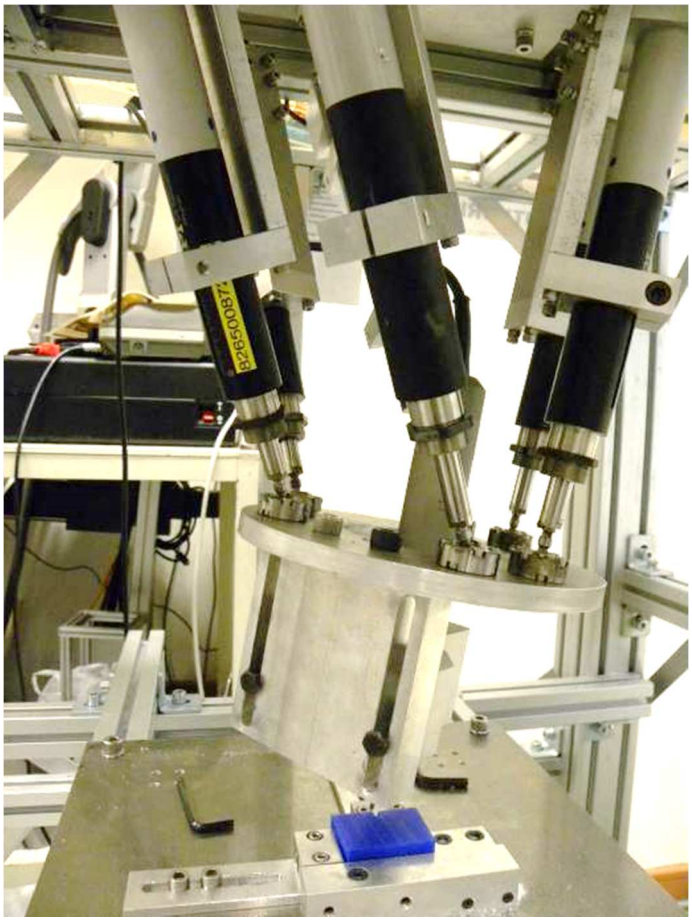


Figure 7.15 The tool-SP in its most tilted configuration

A 2 ½D work-piece, which is the dome shaped part, was designed to verify the error of the calibration process. This verification method includes the identification of error factors from the Stewart platform kinematics and dynamics errors during machining. A CMM with a pre-calibrated touch trigger probe was used to measure points on the machined surface. The CMM was programmed to follow curves on the machined profile, which is generated from the local part coordinates obtained from the vertices of the faces forming the surface of the work-piece as shown in Figure 7.16. Figure 7.17 shows the 3D plot of the measured points on the surface. The mean of

the absolute deviation from the defined work-piece before and after calibration is less than 0.5 mm and 0.05 mm respectively.

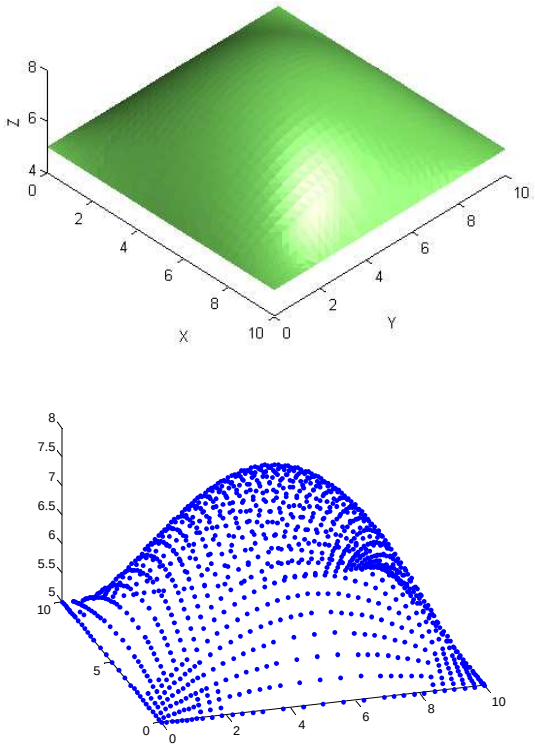
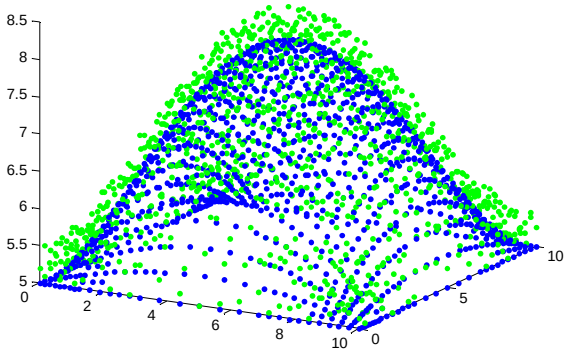
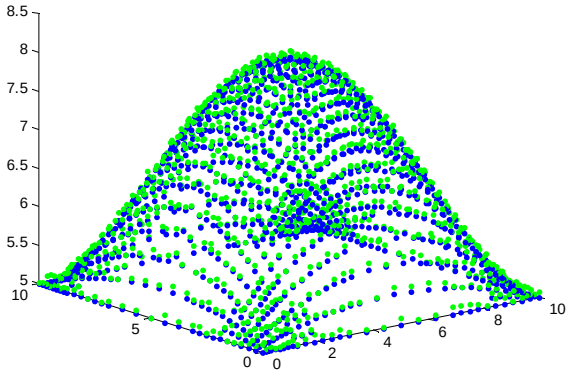


Figure 7.16 Surface model and extracted vertices of the dome shaped test part



(a) Before calibration



(b) After calibration

Figure 7.17 Comparison of surface measurement

7.5 Stewart Platforms Evaluation

Since the emergence of the first Stewart platform based machine tool decades ago, researchers are investigating whether the full potential of the PKM will actually change the machining processes. The promising characteristics that have been observed in Stewart platforms are higher payload-to-weight ratio, noncumulative joint error, higher structural rigidity, and modularity. Nevertheless, the design of these PKMs is the key to bring out these benefits to the machining industry. This researcher has experienced difficulties in finding good components, especially the passive joints of the Stewart platform. In addition, the frame in which the Stewart platforms are built must be rigid to maintain accuracy after calibration. Furthermore, the connection between components of the Stewart platforms creates additional error on top of these errors.

Due to these limitations, the micrometre machining accuracy is difficult to be achieved with the current setup. The tool-SP was built with the best joints in the market, and presently, it has the highest accuracy among the

others. However, it is still below the performance of conventional CNC machines. Finally, the accuracy is not satisfactory because a few main causes: the vibration of the frame structure, tool holder misalignment, and mostly from the spherical joint clearance of the table-SP which produces uncontrollable motion at its end-effector.

In addition, there are a few aspects which are worth mentioning in this evaluation. While it is generally accepted that PKMs have a low workspace-to-footprint ratio, simply adding some automation components could increase the workspace efficiently. For example, an additional rotary table with one or two DOF would increase the tool approach angle of the tool-SP. The rotary table could add more flexibility compared to the table-SP. Since the tool-SP workspace is strongly affected by the tool orientation, the rotary table could replace the orientation capability of the platform, making the job easier and maintain the translational workspace of the tool-SP to maximum at all times. This is similar to the decoupling concept in the serial manipulator, where the hand performs all the translation and the wrist performs all the rotation required by the task. Fortunately, the table-SP can act as a multi-DOF table. Thus, one can see the second SP not as another manipulator, but as an automation component to help the main tool-SP perform its job.

Calibration of a Stewart platform to eliminate its static error is not easy. The calibration of PKMs has predominantly been evaluated in terms of analyses using kinematic-based models. Basically, the errors come from imperfect knowledge of manipulators' parameters. Some problematic error sources that are hard to address include localized heat generation in the

screw/nut system, axis errors, and nonlinear kinematic mapping errors. It is interesting that the calibration issue is also related to stiffness analysis, work-piece placement, and controller development. All of these other issues can affect the accuracy in different ways. Due to its complexity and time-consuming calibration, the introduction of PKM in the manufacturing context needs the development of a fast and automated calibration technique.

The control system of a manipulator is also a crucial issue. The control of a PKM is important when dealing with the effect of leg inertia and dynamics behaviour. What is required for the control system of a PKM is the possibility of introducing a real-time, singularity-free path as well as workspace verification. This should also be done *a priori* in the CAM simulation and NC verification procedures. In addition, it is necessary to check that the trajectory does not exceed the workspace limits. The performance of a PKM varies over its workspace, thus when a tool-path is executed, optimization is needed to assure a smooth machining operation to provide a maximum, stiffness along the machining path between the work-piece and the tool.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

In this thesis, an introduction to the closed-chain mechanism of PKMs has been presented, including its brief history, literature review, applications, and comparison with serial kinematic manipulators. Compared to serial kinematic manipulators, Stewart platforms have the advantages of higher structural stiffness, larger payload capacity and lower friction. The primary shortcoming of Stewart platforms is the limited workspace.

In this research, the development of two Stewart platforms has been presented and they can be applied in various applications, such as machining and positioning. The focus is to control and plan the motion of Stewart platforms to achieve better performance compared to serial counterparts.

Cooperative manipulators consisting of two Stewart platforms has been fabricated and developed in this research. A frame is designed for the two Stewart platforms to be installed, such that one Stewart platform is installed in a normal configuration and a second Stewart platform is installed in an inverted configuration, such that they can be used as a hybrid mechanism to perform a task cooperatively. Various components for calibration and machining purposes are installed and attached on this frame,

such as wire-sensors, camera, cabling compartments, machine tool attachment point, and other peripherals that are needed for the control systems.

Software development in this research includes a graphical user interface which uses a mouse and a joystick as devices to control the motion of the Stewart platforms. In addition, a user can control the movement of the Stewart platforms by providing input parameters in terms of the coordinates in the various coordinate systems, namely, the pose of the moving platform and the joint coordinates of the actuators. Inverse and forward kinematics analyses are included in this control system to observe the motion of the Stewart platforms continuously. The inverse kinematics is found to have a direct solution while the forward kinematics can be solved by using the Newton-Raphson numerical method. A graphical simulation program based on M-file programming and MatLab SimMechanics has been developed and used as a valuable design tool to investigate the effects of geometric parameters and joint constraints on the motion of the Stewart platforms and to provide useful information about the workspace, joint angle, and dynamics of the leg lengths. The graphical simulation program is also used with the user interface to simulate the experiments, check software malfunction and validate the control process.

The approach based on least squares formulation (LSF) is developed to correct the positioning of the moving platform of the Stewart platforms and compensate for the deviations of the kinematic parameters from the nominal values due to manufacturing and assembly errors of the Stewart platforms components. The kinematic error model of the Stewart platforms is

formulated based on a standard parameters set that is accepted in most research literature. The analytical formulas are developed and simulated prior to applying the data collected from sensors. In particular, dial gauges and wire-sensors are used to calibrate the Stewart platforms. Significant error reduction was observed after the calibration process. Furthermore, a PSO search finding measurement for optimum calibration is developed. A calibration work-piece was used to verify the translation movement of the Stewart platform and the improvement of the control software. The calibration results are useful to improve the absolute and relative accuracy and verify the resolution of the Stewart platform.

The machining process using the cooperative manipulators is achieved by using NC codes. Data files containing a series of poses can be used to move the platform in pre-computed trajectories. These trajectories are generated from a conversion of these NC codes using MatLab programming, which could generate sufficient trajectory description of a Stewart platform configuration from an NC file. Three- and five-axis machining processes have been achieved for the cooperative manipulators based on coordinate mapping. The NC file is generated using commercial CAM software, namely MasterCAM in this project, based on 3D models of the work-pieces to be machined.

8.2 Research Contributions

This work has incorporated two fields of research, namely, the field of machine tools, and the subject of parallel kinematic manipulation. Detailed

studies of the proposed cooperative manipulators have led to some generally applicable concepts for hybrid manipulation for tasks that require high stiffness and large workspace. The following points summarize the research achievements in this thesis.

- Design and implementation of the tool-SP and the table-SP.
- Kinematic analysis and control of the Stewart platforms.
- Implementation of user interface for controlling both Stewart platforms.
- Integration of the two Stewart platforms to form the cooperative manipulators to perform machining tasks.
- Conversion of CNC codes from a CAD/CAM package to be used in the cooperative manipulators as a machine tool.
- Analysis of workspace improvement from the single configuration to the extended configuration.
- Development of calibration methods to improve motion accuracy.
- Machining case studies with the proposed cooperative manipulators.

The main contribution of this thesis consists of the experimental verification of the proof of concept of two cooperative PKMs as a machine tool, the proposed method for finding optimum motion planning of two parallel manipulators in performing a single machining task, and the proposed method of accuracy improvement of the Stewart platforms.

8.3 Future Work

New methods and innovative design solutions are needed to solve problems associated with vibration, stiffness, accuracy, calibration and temperature compensation. The benefits of the PKMs can be achieved but they require considerable efforts for their proper application and generalization. Hence, future works can be focused on the following aspects.

1. With some modifications, the algorithms proposed can be used in other applications, such as grinding and polishing. More accurate results can be expected by installing 6-DOF positional or force sensors on the moving platform and using the additional feedback for the control algorithms. Currently, the control system used is based on industrial PID, which controller cards are hard-coded with un-modifiable control loop for each individual leg. If the controller feedback includes information from the sensors on the end-effector, dynamic effects caused by the interaction of the tool and the work-piece can be compensated.
2. The design of the Stewart platform or the synthesis of a new type of PKM can be improved. This should include a dynamic model that could be used in accuracy analysis, dynamic stiffness analysis, and advanced control. In order to obtain good resolution and accuracy, a thorough modelling of the issues could be used for error compensation or redesigning of a new component. Currently, the main problem is due to the backlash of the joints, platform vibration, limited calibration devices, and manual fixation of the Stewart platform base.

3. The mechanical design can be reviewed to increase the present working envelope. The results have shown that the table-SP can increase the workspace of the tool-SP significantly. One option is to redesign the structural arrangement of the joints of the table-SP. An approach would be to combine serial and parallel chain mechanisms to build cooperative manipulators structure with a larger positional workspace. Other arrangements of the kinematics chains can be explored to extend the cooperative manipulators depending on the application requirement.
4. In future, medical applications of Stewart platforms or PKM can be explored. A few research studies on this area have proven that the utilization of PKMs in the medical field is more preferred than serial kinematic manipulators because of their capabilities of handling small devices with higher accuracy and with smaller installation space. A PKM can also be implemented as the end-effector of a serial robot. The development of an effective user interface for conveying the commands of a surgeon to the machine is a key prerequisite for a medical robotic assistant.

BIBLIOGRAPHY

Andreff N., Renaud P., Martinet P., Pierrot F.. (2004). Vision-based kinematic calibration of an H4 parallel mechanism: practical accuracies. Industrial Robot; 31(3): p. 273-83.

Andreff N., Dallej T., Martinet P., (2007). Image-based visual servoing of a Gough-Stewart parallel manipulator using leg observations. International Journal of Robotics Research; 26(7): p. 677-87.

Bhattacharya S., Hatwal H., Ghosh A., (1998). Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators. Mechanism and Machine Theory; 33(7): p. 965-974.

Bonev I.A., Ryu J., (2001). A new approach to orientation workspace analysis of 6-DOF parallel manipulators. Mechanism and Machine Theory; 36(1): p. 15-28.

Bonev I.A., Zlatanov D., Gosselin C.M., (2002). Advantages of the modified Euler angles in the design and control of PKMs. In: Proceeding of the Parallel Kinematic Machines International Conference. Chemnitz, Germany; p. 171-188.

Callegari M., Suardi A. (2003) Hybrid Kinematic Machines for Cooperative Assembly Tasks. In: Proceedings of International Workshop: Multiagent Robotic Systems: Trends and Industrial Applications, Robocup. Padova; 7 July 2003.

Carbone G., Ceccarelli M. (2005). A serial-parallel robotic architecture for surgical tasks. Robotica; 23(3): p. 345-54.

Chanal H., Duc E., Ray P., Hascoët J.Y., (2007). A new approach for the geometrical calibration of parallel kinematics machines tools based on the machining of a dedicated part. International Journal of Machine Tools and Manufacture; 47(7-8): p. 1151-63.

Chai K.S., Young K. (2001) Designing a Stewart platform-based Cooperative System for Large Component Assembly. In: Proceedings of IEEE Conference on Methods and Models in Automation and Robotics. Miedzyzdroje; 28-31 August 2001.

Chen S.-H., Fu L.-C. (2006) The forward kinematics of the 6-6 Stewart platform using extra sensors. In: Proceedings of IEEE Conference on Systems, Man and Cybernetics. Taipei, 8-11 October 2006.

Chen Y., McInroy J.E., Yi Y. (2003). Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance. IEEE Transactions on Robotics and Automation; 19(4): p. 680-91.

Cheok K.C., Overholt J.L., Beck R.R. (1993). Exact methods for determining the kinematics of a Stewart platform using additional displacement sensors. Journal of Robotic Systems; 10(5): p. 689-707.

Chiacchio P., Pierrot F., Sciavicco L., Siciliano B. (1993). Robust design of independent joint controllers with experimentation on a high-speed parallel robot. IEEE Transactions on Industrial Electronics; 40(4): p. 393-403.

Clavel R., Helmer P., Niaritsiry T., Rossopoulos S., Verettas I. (2005) High precision parallel robots for micro-factory applications. In: Proceedings of 2nd International Colloquium, Collaborative Research Centre 562. Braunschweig; 10-11 May 2005. p. 285-296.

Codourey A., Burdet E. (1997) A body-oriented method for finding a linear form of the dynamic Equation of fully parallel robots. In: Proceedings of IEEE Conference on Robotics and Automation. Albuquesrque, USA; 20-25 April 1997. p. 1612-1618.

Cortes, J. Simeon T. (2003) Probabilistic motion planning for parallel mechanisms. In: Proceedings of IEEE Conference on Robotics and Automation. Taipei, Taiwan; 14-19 September 2003. p. 4354-4359.

Dallej T., Hadj-Abdelkader H., Andreff N., Martinet P. (2006a) Kinematic calibration of a Gough-Stewart platform using an omnidirectional camera. In: Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems. Beijing, China; 9-15 October 2006. p. 4666-4671.

Dallej T., Andreff N., Mezouar Y., Martinet P. (2006b) 3D pose visual servoing relieves parallel robot control from joint sensing. In: Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems. Beijing, China; 9-15 October 2006. p. 4291-4296.

Daney D. (2002) Optimal measurement configurations for Gough platform calibration. In: Proceedings of IEEE Conference on Robotics and Automation. Washington, DC, USA; 11-15 May 2002. p. 147-152.

Daney D., Madeline B., Papegay Y. (2005). Choosing measurement poses for robot calibration with the local convergence method and Tabu search. International Journal of Robotics Research; 24(6): p. 501-508.

Daney D., Andreff N., Chabert G., Papegay Y. (2006). Interval method for calibration of parallel robots: Vision-based experiments. Mechanism and Machine Theory; 41(8): p. 929-944.

Dasgupta B., Choudhury P. (1999). A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. Mechanism and Machine Theory; 34(6): p. 801-824.

Dasgupta B., Mruthyunjaya T.S. (2000). Stewart platform manipulator: A review. Mechanism and Machine Theory; 35(1): p. 15-40.

Dietmaier, P. (1998) The Stewart-Gough Platform of General Geometry can have 40 real postures. In: Proceedings of 6th International Symposium on Advances in Robot Kinematics: Analysis and Control. Strobl, Austria; 29 June - 4 July 1998. p. 7-16.

Du Plessis L.J., Snyman J.A. (2001). A numerical method for the determination of dextrous workspaces of Gough-Stewart platforms. International Journal for Numerical Methods in Engineering; 52(4): p. 345-69.

El-Khasawnath BS, Ferreira PM, (1999). Computation of stiffness and stiffness bounds for parallel link manipulators. International Journal of Machine Tools and Manufacturing; 39(2): p. 321–342.

Fassi I., Wiens G.J. (2000). Multiaxis machining: PKMs and traditional machining centres. Journal of Manufacturing Processes; 2(1): p. 1-14.

Fichter E.F. (1986). A Stewart-Platform based manipulator: general theory and practical construction. International Journal of Robotic Research; 5(2): p. 157-182.

Gallardo, J., Rico J.M., Frisoli A., Checcacci D., Bergamasco M. (2003). Dynamics of parallel manipulators by means of screw theory. Mechanism and Machine Theory; 38(11): p. 1113-1131.

Geng, Z., Haynes L.S., Lee J.D., Carroll R.L. (1992). On the dynamic model and kinematic analysis of a class of Stewart platforms. Robotics and Autonomous Systems; 9(4): p. 237-54.

Geng, Z., Haynes L.S.(1994) An Effective Kinematic Calibration Method for Stewart platforms. In: Proceedings of International Symposium on Robotics and Manufacturing. Hawai; 15-17 August 1994. p. 87-92.

Geldart M., Webb P., Larsson H., Backstrom M., Gindy N., Rask K., (2003). A direct comparison of the machining performance of a variax 5 axis parallel kinetic machining centre with conventional 3 and 5 axis machine tools. International Journal of Machine Tools and Manufacture; 43(11): p. 1107-1116.

Golub G. H., Van Loan, C. F. Matrix Computations, Baltimore: The Johns Hopkins University Press, 1983.

Gosselin, C., Angeles J. (1990a). Singularity analysis of closed-loop kinematic chains. IEEE Transactions on Robotics and Automation; 6(3): p. 281-290.

Gosselin, C. (1990b). Determination of the Workspace of 6-DOF Parallel Manipulators. Journal of Mechanical Design; 112(3): p. 331-336.

Gosselin, C., Angeles J. (1990c). Kinematic inversion of parallel manipulators in the presence of incompletely specified tasks. Journal of Mechanical Design; 112(4): p. 494-500.

Gough, V.E. Whitehall S.G. (1962) Universal tire test machine. In: Proceedings of the 9th FISITA World Automotive Congress. London; 1962, p. 117-137.

Han C., Kim J., Kim J., Park F.C., (2002). Kinematic sensitivity analysis of the 3-UPU parallel mechanism. Mechanism and Machine Theory; 37(8): p. 787-798.

Harib K.H, Srinivasan K. (2003). Kinematic and dynamic analysis of Stewart platform-based machine tool structures. Robotica; 21(5): p. 541-554.

Harib K.H., Sharif Ullah A.M.M., Hammami A. (2007). A hexapod-based machine tool with hybrid structure: Kinematic analysis and trajectory planning. International Journal of Machine Tools and Manufacture; 47(9): p. 1426-32.

Harris, D.M.J. (1995) Parallel-linkage robot coordinate transformation horugh screw theory. In: Proceedings of 9th World Congress on the Theory of Machines and Mechanism. Milan; 30 August-2 Sept 1995. p. 1565-1568.

Huang C.-I., Chang C.-F.. Yu M.-Y., Fu L.-C. (2004) Sliding-mode tracking control of the Stewart platform. In: Proceedings of 5th Asian Control Conference. Melbourne, Australia; 20-23 July 2004. p. 562-569.

Huang C.-I., Fu L.-C. (2004) Adaptive backstepping tracking control of the Stewart platform. In: Proceedings of 43rd IEEE Conference on

Decision and Control. Nassau, Bahamas; 14-17 December 2004. p. 5228-5223.

Huang T., Wang J., Gosselin C.M., Whitehouse D.J. (1999). Determination of closed form solution to the 2-D orientation workspace of Gough-Stewart parallel manipulators. IEEE Transactions on Robotics and Automation; 15(6): p. 1121-5.

Huang T., Whitehouse D.J. (2000). A Simple yet Effective Approach for Error Compensation of a Tripod-Based Parallel Kinematic Machine. CIRP Annals - Manufacturing Technology; 49(1): p. 285-288.

Huang T., Chetwynd D.G., Whitehouse D.J., Wang J. (2005). A general and novel approach for parameter identification of 6-DOF parallel kinematic machines. Mechanism and Machine Theory; 40(2): p. 219-239.

Hunt K.H., Kinematic Geometry of Mechanism, Oxford: Clarendon Press, 1978.

Kennedy J., Eberhart R. (1995) Particle swarm optimization. In: Proceedings of IEEE Conference on Neural Networks. Perth, Australia; 27 November-1 December 1995. p. 1942-1948.

Kim S., S. Lee. (1992) Cartesian space dynamic model of serial-parallel manipulator systems and their dynamic performance evaluation. In Proceedings of 31st IEEE Conference on Decision and Control. Tucson, AZ, USA; 16-18 December 1992. p. 327-328.

Kock S., Schumacher W. (2000) A mixed elastic and rigid-body dynamic model of an actuation redundant parallel robot with high-reduction

gears. In: Proceedings of IEEE Conference on Robotics and Automation. San Francisco, CA, USA; 24-28 April 2000. p. 1918-1923

Lallemand J.P., Goudali A., Zeghloul S. (1997). The 6-DOF 2-Delta parallel robot. Robotica; 15: p. 407-16.

Li C.-G., Ding H.-S., Wu P.-D. (2003) Application of MRAC to a 6-dof parallel machine tool. In: Proceedings of IEEE Conference on Machine Learning and Cybernetics. Xi'an, China; 2-5 November 2003. p. 2164-2167.

Liu G.F., Wu Y.L., Wu X.Z., Kuen Y.Y., Li Z.X. (2001) Analysis and control of redundant parallel manipulators. In: Proceeding of IEEE Conference on Robotics and Automation. Seoul, Korea; 21-26 May 2001. p. 3748-3754.

Liu, K., Lewis F., Lebret G., Taylor D. (1993). The singularities and dynamics of a Stewart platform manipulator. Journal of Intelligent and Robotic Systems; 8(3): p. 287-308.

Ma O., Angeles J. (1991a) Architecture singularities of platform manipulators. In: Proceedings of IEEE Conference on Robotics and Automation. Sacramento, CA, USA; 9-11 April 1991 p. 1542-1547.

Ma O., Angeles J. (1991b) Optimum Architecture Design of Platform Manipulators. In: Proceedings of the 5th International Conference on Advanced Robotics. 19-22 June 1991. p. 1130-1135.

Marquet F., Company O., Krut S., Pierrot F. (2002) Enhancing parallel robots accuracy with redundant sensors. In: Proceedings of IEEE Conference

on Robotics and Automation. Washington, DC, USA; 11-15 May 2002. p. 4114-4119.

Masory, O., Wang J., (1995). Workspace evaluation of Stewart platforms. Journal of Advanced Robotics; 9(4): p. 443-61.

Masory O., Wang J., Zhuang H. (1997) Kinematic modelling and calibration of a Stewart platform. Journal of Advanced Robotics; 11(5): p. 519-541.

Meng J., Li Z. (2005) A general approach for accuracy analysis of parallel manipulators with joint clearance. In: Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems. Canada; 2-6 August 2005 p. 2468-2473.

Merlet J.P. (1992) On the infinitesimal motion of a parallel manipulator in singular configurations. In: Proceedings of IEEE Conference on Robotics and Automation. Nice, France; 12-14 May 1992. p. 320-325.

Merlet J.P. (1994) Trajectory verification of parallel manipulators in the workspace. In: Proceedings of IEEE Conference on Robotics and Automation. San Diego, CA, USA; 8-13 May 1994. p. 2166-2171.

Merlet J.P. (1999) Parallel Robots: Open Problems. In: Proceedings of 9th International Symposium of Robotics Research. Snowbird, Utah, USA; 9-12 October 1999.

Merlet J.P., Perng M-W., Daney. D. (2000) Optimal Trajectory Planning of a 5-axis Machine-Tool based on a 6-axis parallel Manipulator. In: Porceedings of 7th International Symposium on Advances in Robot Kinematics. Piran, Slovenia; June 2000. p. 315-322.

Merlet J.P. (2002) Still a long way to go on the road for parallel mechanisms. In: ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Montreal, Quebec, Canada; 29 September-2 October 2002.

Merlet J.P., Parallel Robot. Solid Mechanics and Its Application., Dordrecht: Springer, 2006.

Merlet J.P. (2007). A formal-numerical approach for robust in-workspace singularity detection. IEEE Transactions on Robotics; 23(3): p. 393-402.

Nahvi A., Hollerbach J.M. (1996) The noise amplification index for optimal pose selection in robot calibration. In: Proceedings of IEEE Conference on Robotics and Automation. Minneapolis, MN, USA; 22-28 April 1996. p. 647-654.

Neugebauer R., Wieland F., Ihlenfeldt S. (2000). Comparison of Parallel Structure Concepts for Five-Axis Machining. Journal of Manufacturing Processes; 2(1): p. 59-66.

Neumann K.E., (2006). Exechon Concept. In: The 5th Chemnitz Parallel Kinematics Seminar. Allemagne; p.787-802.

Nguyen C.C., Pooran F.J. (1989). Dynamic analysis of a 6 DOF CKCM robot end-effector for dual-arm telerobot systems. Robotics and Autonomous Systems; 5(4): p. 377-94.

Nguyen C.C., Zhou Z.-L., Antrazi S.S., Campbell C.E., Jr. (1991) Efficient computation of forward kinematics and Jacobian matrix of a Stewart

platform-based manipulator. In: Proceedings of IEEE SoutheastCon 1991. Williamsburg, VA, USA; 7-10 April 1991. p. 869-874.

Niaritsiry F.-T., Fazenda N., Clavel R. (2004) Study of the source of inaccuracy of a 3 DOF flexure hinge-based parallel manipulator. In: Proceedings of IEEE Conference on Robotics and Automation. New Orleans; 26 April-1 May 2004. p. 4091-4096.

Parenti-Castelli V., Di Gregorio R. (1995). Determination of the actual configuration of the general Stewart platform using only one additional displacement sensor. Journal of Mechanical Design; 121(1):p. 21-25.

Parenti-Castelli V., Di Gregorio R. (2000). A New Algorithm Based on Two Extra-Sensors for Real-Time Computation of the Actual Configuration of the Generalized Stewart-Gough Manipulator. Journal of Mechanical Design; 122(3): p. 294-298.

Parikh P.J., Lam S.S.Y. (2005). A hybrid strategy to solve the forward kinematics problem in parallel manipulators. IEEE Transactions on Robotics; 21(1): p. 18-25.

Patel A.J., Ehmann K.F. (1997). Volumetric error analysis of a Stewart platform-based machine tool. CIRP Annals - Manufacturing Technology; 46(1): p. 287-290.

Patel A. J., Ehmann K. F., (2000). Calibration of a hexapod machine tool using a redundant leg. International Journal of Machine Tools and Manufacture; 40(4): p. 489-512.

Pierrot F. (2002) Parallel mechanism and redundancy. In: Proceedings of 1st International Colloquium, Collaborative Research Centre 562. Braunschweig; 29-30 May 2002. p. 261-277.

Pritschow G., Eppler C., Garber T. (2002). Influence of the dynamic stiffness on the accuracy of PKM. In: Proceedings of the 3rd Chemnitzer Parallelkinematik Seminar. Chemnitz, Germany; May 2002. p. 313-333.

Pugazhenthi S., Nagarajan T., Singaperumal M. (2002). Optimal trajectory planning for a hexapod machine tool during contour machining. Proceedings of the Institution of Mechanical Engineer, Part C: Journal of Mechanical Engineering Science; 216(12):p. 1247-1257.

Reboulet C., Berthomieu T. (1991) Dynamic models of a six degree of freedom parallel manipulators. In: Proceedings of the 5th International Conference on Advanced Robotics. 19-22 June 1991. p. 1153-1157.

Refaat S., Herve J.M., Nahavandi S., Trinh H., (2007). Two-mode overconstrained three-dofs rotational-translational linear-motor-based parallel-kinematics mechanism for machine tool applications. Robotica; 25(4): p. 461-466.

Renaud P., Andreff N., Lavest J.-M., Dhome M. (2006). Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology. IEEE Transactions on Robotics; 22(1): p. 12-22.

Ropponen T., Arai T. (1995) Accuracy analysis of a modified Stewart platform manipulator. In: Proceedings of IEEE International

Conference on Robotics and Automation. Nagoya, Aichi, Japan; 21-27 May 1995. p. 521-525.

Stewart D. (1965). A platform with six degrees of freedom. In: Proceedings of the Institution of Mechanical Engineers; 180(15): p. 371-386.

St-Onge B.M., Gosselin C.M. (2000). Singularity analysis and representation of the general Gough-Stewart platform. International Journal of Robotics Research; 19(3): p. 271-88.

Stoughton R. S., Arai T. (1993). A Modified Stewart Platform Manipulator with Improved Dexterity. IEEE Transactions on Robotics and Automation; 9(2):p. 166-73.

Schwaar M., Jaehnert T., Ihlenfeldt S., (2002). Mechatronic design, experimental property analysis and machining strategies for a 5-strut-PKM. In: The 3rd Chemnitz Parallel Kinematics Seminar: Development Methods and Application Experience of Parallel Kinematics. Verlag Wissenschaftliche Scripten, Zwickau; p. 671-681.

Takeda Y., Shen G., Funabashi H. (2004). A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series. Journal of Mechanical Design; 126(5): p. 856-65.

Tanaka W., Arai T., Inoue K., Takubo T., Park C.S. (2006) Calibration method for parallel mechanism using micro grid pattern. In: Proceeding of IEEE Conference on Robotics and Automation. Orlando, FL, USA; 15-19 May 2006. p. 763-768.

Tanev T.K. (2000). Kinematics of a hybrid (parallel-serial) robot manipulator. Mechanism and Machine Theory; 35(9): p. 1183-1196.

Tarokh M. (2007) Real time forward kinematics solutions for general Stewart platforms. In: Proceeding of IEEE Conference on Robotics and Automation. Rome, Italy; 10-14 April 2007. p. 901-906.

Terrier M., Giménez M., Hascoët J.Y., (2005). Verne - a five-axis parallel kinematics milling machine. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture; 219(3): p. 327-336.

Tischler C.R., Samuel A.E. (1998) Predicting the Slop of In-Series/Parallel Manipulators caused by Joint Clearances. In: Proceedings of 6th International Symposium on Advances in Robot Kinematics: Analysis and Control. Strobl, Austria; 29 June - 4 July 1998. p. 227-236.

Tlusty J., Ziegert J.C., Ridgeway S. (1999). Fundamental comparison of the use of serial and parallel kinematics for machines tools. CIRP Annals - Manufacturing Technology; 48(1): p. 351-356.

Tlusty J., Ziegert J.C., Ridgeway S. (2000). A Comparison of Stiffness Characteristics of Serial and Parallel Machine Tools. Journal of Manufacturing Processes; 2(1): p. 67-76.

Tsai L.-W. (2000). Solving the Inverse Dynamics of a Stewart-Gough Manipulator by the Principle of Virtual Work. Journal of Mechanical Design; 122(1): p. 3-9.

Tsai L.-W., Joshi S. (2002). Kinematic Analysis of 3-DOF Position Mechanisms for Use in Hybrid Kinematic Machines. Journal of Mechanical Design; 124(2): p. 245-253.

Voglewede P.A., Ebert-Uphoff I., (2005). Overarching framework for measuring closeness to singularities of parallel manipulators. IEEE Transactions on Robotics; 21(6): p. 1037-1045.

Wampler C.W., Hollerbach J.M., Arai T. (1995). An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. IEEE Transactions on Robotics and Automation; 11(5): p. 710-724.

Wang H., Xue C., Gruver W.A. (1995) Neural network control of a parallel robot. In: Proceedings of IEEE Conference on Systems, Man and Cybernetics. Vancouver, BC, Canada; 22-25 October 1995. p. 2934-2938.

Wang J., Masory O. (1995) On the accuracy of Stewart Platform – Part I the effect of manufacturing tolerances. In: Proceedings of IEEE International Conference on Robotics and Automation. Nagoya, Aichi, Japan; 2-6 May 1995. p. 114 – 120.

Wang J., Gosselin C.M. (1998). A New Approach for the Dynamic Analysis of Parallel Manipulators. Multibody System Dynamics; 2(3): p. 317-334.

Wang J., Gosselin C.M. (2004). Kinematic analysis and design of kinematically redundant parallel mechanisms. Journal of Mechanical Design; 126(1): p. 109-18.

Wang, S.-M., Ehmann K.F. (2002). Error Model and Accuracy Analysis of a Six-DOF Stewart platform. Journal of Manufacturing Science and Engineering; 124(2): p. 286-295.

Wang, Y. (2006) An incremental method for forward kinematics of parallel manipulators. In: Proceedings of IEEE Conference on Robotics, Automation and Mechatronics. Bangkok, Thailand; 1-3 June 2006. p. 1-5.

Wang, Z., Liu W., Lei Y. (2001). A study on workspace, boundary workspace analysis and work-piece positioning for parallel machine tools. Mechanism and Machine Theory; 36(5): p. 605-622.

Weck M., Staimer D., (2002). Parallel Kinematic Machine Tools – Current State and Future Potentials. CIRP Annals - Manufacturing Technology; 51(2): p. 671-683.

Wohlhart K. (1999). Degrees of shakiness. Mechanism and Machine Theory; 34(7): p. 1103-1126.

Yurt S.N., Őzkol I., Kaya M.O., Haciyev C., (2002). Optimization of the PD coefficient in a flight simulator control via genetic algorithms. Aircraft Engineering and Aerospace Technology; 74(2): p. 147-153.

Zhang D., Wang L., Lang S.Y.T. (2005). Parallel Kinematic Machines: Design, Analysis and Simulation in an Integrated Virtual Environment. Journal of Mechanical Design; 127(4): p. 580-588.

Zheng G., Haynes L.S. (1993). Dynamic control of a parallel link manipulator using a CMAC neural network. Computers and Electrical Engineering; 19(4): p. 265-76.

Zheng X.Z., Bin H.Z., Luo Y.G. (2004). Kinematic analysis of a hybrid serial-parallel manipulator. International Journal of Advanced Manufacturing Technology; 23(11-12): p. 925-30.

Zhuang H., Roth Z.S. (1993). Method for kinematic calibration of Stewart platforms. Journal of Robotic Systems; 10(3): p. 391-405.

Zhuang H., Yan J., and Masory O. (1998). Calibration of Stewart Platform and other parallel manipulators by minimizing inverse kinematic residuals. Journal of Robotic Systems; 15(7): p. 396-406.

Zlatanov D., Bonev I.A., Gosselin C.M., (2002). Constraints singularities of parallel mechanisms. In: Proceedings of IEEE Conference on Robotics and Automation. p. 496-502.

Zuo A., Wu Q.M.J., Gruver W.A. (2002) Stereo vision guided control of a Stewart platform. In: Proceedings of IEEE International Symposium on Intelligent Control. Vancouver, BC, Canada; 30 October 2002. p. 125-130.

PUBLICATIONS FROM THIS RESEARCH

1. Saputra. V.S., Ong S.K., Nee A.Y.C., Development and Trajectory Planning of a Hybrid Serial-Parallel Manipulator, Asian International Journal of Science and Technology in Production and Manufacturing Engineering, 2009, 2(1): p. 99-115.
2. Saputra. V.S., Ong S.K., Nee A.Y.C., A PSO algorithm for mapping the workspace boundary of parallel manipulators, In Proceedings of International Conference on Robotics and Automation, Anchorage, Alaska, US; 3-7 May 2010, p. 4691-4696.
3. Saputra. V.S., Ong S.K., Nee A.Y.C., A Swarm Optimization Approach for Solving Workspace Determination of Parallel Manipulators, Robotica, July 2012, under review.
4. Saputra. V.S., Ong S.K., Nee A.Y.C., Optimum Calibration of a Parallel Kinematic Manipulator (PKM) using Digital Indicators, Journal of Intelligent and Robotic Systems, October 2012, under review.